

DOCUMENT RESUME

ED 235 807

IR 050 464

AUTHOR Meadow, Charles T.
TITLE A Study of User Adaptation to an Interactive Information Retrieval System. Research Report.
INSTITUTION OCLC Online Computer Library Center, Inc., Dublin, Ohio.
REPORT NO OCLC/OPR/RR-83/6
PUB DATE 15 Jul 83
NOTE 119p.
PUB TYPE Reports - Research/Technical (143) -- Tests/Evaluation Instruments (160)

EDRS PRICE MF01/PC05 Plus Postage.
DESCRIPTORS Attitude Measures; *Databases; Foreign Countries; *Information Retrieval; *Online Systems; *Performance; Questionnaires; *Relevance (Information Retrieval)

IDENTIFIERS *Command Language; England; *Online Search Skills

ABSTRACT

A study was undertaken in 1980-81 to investigate the process of user adaptation to an unfamiliar interactive database search system. Adaptation was defined as performance on, knowledge of, and attitude toward the system being used. The experiment tested 44 students and staff members at the University of Sheffield and Sheffield City Polytechnic in England. Users had three levels of prior computer experience. Each user was trained in one of three command languages with increasing levels of complexity and logical power, and then performed 20 searches of a database consisting of some 580 condensations of news articles appearing in British national newspapers between October 1980 and March 1981. Measures of searcher performance such as recall, precision, and self-evaluation were computed, and attitude questionnaires were also administered. It was found that performance increased with experience and that users appeared to perform best with a language of power and complexity commensurate with their own skills at the time of use; i.e., low power/complexity for inexperienced, high power/complexity for experienced users. The report includes an 18-item bibliography, the text of an online help file, a list of the test questions searched, and sample training handouts and questionnaires. (Author/ESR)

* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *

ED235807

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

Report Number: OCLC/OPR/RR-83/6
Date: 1983 July 15

- ☒ This document has been reproduced as received from the person or organization originating it.
- ☐ Minor changes have been made to improve reproduction quality.
- Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

Research Report Prepared for OCLC

on

A Study of User Adaptation to an Interactive Information Retrieval System

by

Charles T. Meadow

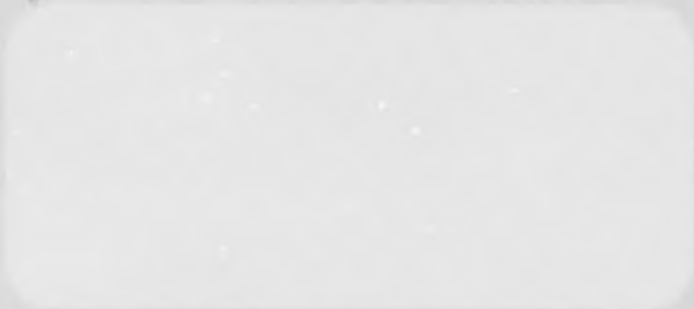
OCLC Online Computer Library Center, Inc.
Office of Planning and Research
6565 Frantz Road
Dublin, OH 43017-0702

"PERMISSION TO REPRODUCE THIS
MATERIAL HAS BEEN GRANTED BY

Cray Little

TO THE EDUCATIONAL RESOURCES
INFORMATION CENTER (ERIC)."

IR050464



OCLC is a not-for-profit computer library service and research organization based in Dublin, Ohio. The Center operates an international network that libraries use to acquire and catalog books, order custom-printed catalog cards, arrange interlibrary loans, and maintain location information on library materials.

All rights reserved; however, individuals may quote from or reproduce portions of the text for noncommercial, educational, or private study or research. Full credit must be given to both the author(s) and the Office of Research, OCLC Online Computer Library Center, Inc., Dublin, Ohio.

RESEARCH REPORT SERIES

Through the Research Report Series, OCLC research project results are made available to the public. OCLC Research Reports are published and distributed by OCLC. In addition, the reports are deposited with ERIC, which makes them available in microfiche as well as paper copy. Until Research Reports are available from ERIC (usually six to nine months after publication), OCLC will supply copies to persons and institutions requesting them. Due to the increasing demand for these reports and increasing costs to produce them, OCLC charges requestors a nominal fee. The OCLC price or ERIC number is noted below for each report.

Subject Heading Patterns in OCLC Monographic Records

by E. O'Neill and R. Aluri

Report Number: OCLC/RDD/RR-79/1; ERIC ED 183167; 96 pages

An Overview of a Proposed Monitoring Facility for the Large-scale, Network-based OCLC Online System

by W. Dominick, D. Penniman, and J. Rush

Report Number: OCLC/OPR/RR-80/1; ERIC ED 186042; 30 pages

Analytical Review of Catalog Use Studies

by K. Markey

Report Number: OCLC/OPR/RR-80/2; ERIC ED 186041; 66 pages

A Method for Correcting Typographical Errors in Subject Headings in OCLC Records

by E. O'Neill and R. Aluri

Report Number: OCLC/OPR/RR-80/3; ERIC ED 215680; 34 pages

Field, Subfield, and Indicator Statistics in OCLC Bibliographic Records

by T. B. Hickey

Report Number: OCLC/OPR/RR-81/1; ERIC ED 202484; 156 pages

Development of a Probabilistic Author Search and Matching Technique for Retrieval and Creation of Bibliographic Records

by T. B. Hickey

Report Number: OCLC/OPR/RR-81/2; ERIC ED 206286; 48 pages

Survey of Online Searching Instruction in Schools of Library and Information Science

by C. H. Fenichel and S. P. Harter

Report Number: OCLC/DD/RR-81/3; ERIC ED 210038; 108 pages

The American Public Library

by D. Sager

Report Number: OCLC/OPR/RR-82/1; ERIC ED 221222; 58 pages

The Process of Subject Searching in the Library Catalog: Final Report of
the Subject Access Research Project

by K. Markey

Report Number: OCLC/OPR/RR-83/1

OCLC Price: \$7.00; 152 pages

Current Utilization of Online Catalogs: Transaction Log Analysis;
Final Report to the Council on Library Resources, Volume I

by J. E. Tolle

Report Number: OCLC/OPR/RR-83/2

OCLC Price: \$6.50; 128 pages

Online Catalog Use: Results of Surveys and Focus Group Interviews in
Several Libraries;

Final Report to the Council on Library Resources, Volume II

by K. Markey

Report Number: OCLC/OPR/RR-83/3

OCLC Price: \$12.50; 284 pages

A Comprehensive Study of Online Public Access Catalogs:

An Overview and Application of Findings;

Final Report to the Council on Library Resources, Volume III

by N. K. Kaske and N. P. Sanders

Report Number: OCLC/OPR/RR-83/4

OCLC Price: \$5.50; 102 pages

Multidimensional Mapping of Libraries Based on Shared Holdings in the
OCLC Online Union Catalog

by W. E. McGrath and T. B. Hickey

Report Number: OCLC/OPR/RR-83/5

OCLC Price: \$3.50; 54 pages

A Study of User Adaptation to an Interactive Information Retrieval System

by C. T. Meadow

Report Number: OCLC/OPR/RR-83/6

OCLC Price: \$7.00; 120 pages

OCLC LIBRARY, INFORMATION, AND COMPUTER SCIENCE SERIES

The OCLC Library, Information, and Computer Science Series of monographs is another vehicle through which the OCLC Office of Research communicates the results of significant research investigations and discusses issues of major importance. The first monograph was published in June 1982; the second in June 1983. The third monograph, an analysis of the number of terminals needed to support an online public access catalog, will be available in fall 1983. Additional monographs are planned for publication in 1983-84.

[1] Online Public Access Catalogs: The User Interface

by C. R. Hildreth

Product Number: 801

Price: \$18.00; 280 pages; softbound

2. Public Library Administrators' Planning Guide to Automation

by D. J. Sager

Product Number: 802

Price: \$12.50; 144 pages; softbound

ORDERING INFORMATION

Orders for OCLC publications must be prepaid, unless to be shipped and billed to an OCLC member.

Prepaid orders--send check and order to OCLC, DEPT BOX ONB, COLUMBUS, OHIO 43265.

Orders to be billed--send order, with your OCLC symbol and network indicated, to OCLC DOCUMENTATION DEPT, 6565 FRANTZ ROAD, DUBLIN, OH 43017-0702.

OCLC publications with ERIC numbers assigned should be ordered from ERIC DOCUMENT REPRODUCTION SERVICE, PO BOX 190, ARLINGTON, VA 22210.

ACKNOWLEDGMENTS

The experiment reported herein was performed while I was on sabbatical leave from Drexel University and was supported in part by Drexel and in part by OCLC under a grant designating me as an OCLC Special Scholar. The work was performed at the Postgraduate School of Librarianship and Information Science (PGSLIS, now the Department of Information Studies) of the University of Sheffield, Sheffield, England. The idea for the study grew out of an earlier project of mine, called Individualized Instruction for Data Access (IIDA) (Meadow, Hewett, and Aversa 1982a, 1982b). In that project, my objective was to develop and study a computer system that served as an intermediary in assisting users to perform online bibliographic searches. I became acutely aware of the lack of precise measures of user performance. That suggested the current study, in which the feasibility of measurement was a prime objective.

Measurement implies knowing what and how to measure, and in our field, we are still groping with these things. I hope this report will be received as a modest contribution to the development of more accurate measurement methods.

The project took almost a full year and involved many people whom I would like to thank publicly. They are:

Mrs. Mary Louise Meadow, who designed and built the database and training program, administered the training, and helped to run the experiments. She still does not realize the magnitude of her contribution.

Mr. Fred Kilgour (founder and former president of OCLC), Dr. David Penniman, and Dr. Neal Kaske of OCLC for their foresight in supporting research in general, and their encouragement of my particular effort.

Dr. Fabio Corzo, of the Universidad Francisco de P. Santander, Colombia, who, while a doctoral candidate at the University of Sheffield, did most of the computer programming for the search system.

Dr. Stanley Wardle and Mr. Wilf Webster, University of Sheffield Computing Services Department, for many services and courtesies in connection with my use of their computer.

Mrs. Hetty Harris and Mrs. Felicity Lundu, who helped as attendants in the user experiments.

Dr. Peter Willett and Messrs. John Barnard and Stephen Welford, of PGSLIS, for frequent advice on computer use.

The rest of the PGSLIS staff, whom I badgered and often consulted.

Students and staff of the University of Sheffield and the Sheffield City Polytechnic, who served as subjects or who helped me to find subjects.

Finally, a very special appreciation to Prof. Michael F. Lynch, Mr. Norman Roberts, and Prof. Wilf Saunders (who each were at different times Director or Chairman of the PGS LIS) for inviting me to their school, for providing much patient help of every sort, and, most of all, for extending their friendship in a way that made our stay in Sheffield not only memorable, but also a permanent part of our lives.

Charles T. Meadow
Philadelphia, 1982

ABSTRACT

This is the report of a study undertaken in 1980-81 to investigate the process of user adaptation to an interactive database search system that is new to them. Adaptation was deemed a broader concept than learning in that the former included performance on, knowledge of, and attitude toward the systems being used. Our objective was to discover trends and to explore measures of adaptation. One of the significant conclusions is that users appear to perform best with a language of power and complexity commensurate with their own skills at the time of use: low power/complexity for inexperienced, high power/complexity for experienced users.

CONTENTS

ACKNOWLEDGMENTS, vii

ABSTRACT, ix

TABLES, xiii

1.0 INTRODUCTION, 1

- 1.1 Background, 2
- 1.2 A Conjecture on Language Learning, 3
- 1.3 Objectives and Assumptions, 4
- 1.4 Summary of Results, 6

2.0 DESIGN OF THE EXPERIMENT, 7

- 2.1 The Database, 7
- 2.2 The LURES Search System, 8
 - 2.2.1 General Description, 8
 - 2.2.2 Language 1, 9
 - 2.2.3 Language 2, 10
 - 2.2.4 Language 3, 11
- 2.3 Experimental Subjects, 12
 - 2.3.1 Recruiting, 12
 - 2.3.2 Training, 13
- 2.4 Questions, 14
- 2.5 Scheduling, 15
- 2.6 Pilot Project, 15
- 2.7 Conduct of the Experiment, 16
 - 2.7.1 Arrangements for Searching, 16
 - 2.7.2 Problems, 20
- 2.8 Measures, 22
 - 2.8.1 Subject Background and Environment, 22
 - 2.8.2 Individual Performance Measures--Process, 23
 - 2.8.3 Individual Performance Measures--Outcome, 24
 - 2.8.4 Attitude Toward the System, 25
 - 2.8.5 Composite Measures, 26

3.0 RESULTS, 27

- 3.1 Frequency Distributions and Correlations Among Basic Variables, 27
 - 3.1.1 Variables Related to Process Performance, 27
 - 3.1.2 Variables Related to Outcome, 33
 - 3.1.3 Overall Performance, PERF, 37
 - 3.1.4 Attitude Measures, 37
 - 3.1.5 Question Analysis, 40

3.2	Interrelationships,	41
3.2.1	Question Analysis,	41
3.2.2	Attitude Analysis,	45
3.2.3	Performance Analysis--Process,	48
3.2.4	Performance Analysis--Outcome,	53
3.2.5	Overall Composite Performance,	58
3.2.6	Correlation of Attitude with Performance,	59
3.2.7	Performance Corrected for Question Difficulty,	59
4.0	OVERALL CONCLUSIONS,	63
4.1	Restatement of Assumptions,	63
4.2	Conclusions,	64
	REFERENCES,	65
	APPENDIX A. Reference File,	67
	APPENDIX B. Test Questions,	88
	APPENDIX C. Training Handout,	96
	APPENDIX D. End of Search Questions,	102
	APPENDIX E. End of Session Questions,	103

TABLES

1. Experience Characteristics of the Sample, 13
2. Frequency Distribution for CD, 27
3. Frequency Distribution for ER, 28
4. Frequency Distribution for PA, 29
5. Frequency Distribution for SP, 30
6. Frequency Distribution for PR, 31
7. Frequency Distribution for CP, 32
8. Frequency Distribution for RE, 33
9. Frequency Distribution for RV, 34
10. Frequency Distribution for SO, 35
11. Frequency Distribution for CO, 36
12. Frequency Distribution for PERF, 37
13. Frequency Distribution for AC, 37
14. Frequency Distribution for AD, 38
15. Frequency Distribution for AS, 38
16. Frequency Distribution for AT, 39
17. Frequency Distribution of AT as a Function of IX, 44
18. Frequency Distribution of AD as a Function of IX, 44
19. AT as a Function of Prior Experience (PX), 45
20. AT as a Function of Interface Complexity (IC), 46
21. Attitude by PX and IC, 46
22. AT as a Function of PX, IC, and IX (or Session Number), 46
23. Frequency Distribution of CP by Session, 51
24. CP by PX and IC, 52
25. CP as a Function of PX, IC, and IX (or Session Number), 52
26. CO by PX and IC, 56
27. CO as a Function of PX, IC, and IX (or Session Number), 56

1. INTRODUCTION

This study is concerned with the behavior of users of a database search system from their first exposure to it until they have acquired some mastery of it. In general, the study and its results are presumed applicable to users of artificial languages for instructing a computer in problem-solving situations. Essentially, its basis is the assumption that users of a database search system--or any other artificial language-driven system--go through a period of adaptation when they first begin using the system. In this period, they (1) learn to use the system at a rate which is a function of their prior computer-related experience and the complexity of the system and (2) change their attitude toward the system as a function of their degree of mastery. The user's attitude may decrease (i.e., become negative or less positive than it was previously) after the language is learned if the logical power of the language is less than is needed to solve the kinds of problems that the user's experience enables him to undertake. A user's attitude would probably increase, however, with success at increasingly difficult tasks.

There are two somewhat separate but highly interrelated aspects of the "system"--its functions (what it is capable of doing) and its language (through which the user communicates to the system). We stress the word "to" because the language of talking to a computer is not necessarily the same as that in which the computer talks back to the user. For simplicity, we shall use the term "language" in this report because that is the window through which most users view the information system, i.e., they do not see and are not taught the difference between the language and the logic of the system. As a single example of the difference, it is possible to develop a subset of a programming language, for training beginners, that does not include either subroutines or array notation. As far as the users of the subset are concerned, these logical features do not exist, but the computer and the full language can, of course, handle them. Such differences between language and system are used in this study, but it is the language that is important here.

That there is a learning period during which a user's performance level gradually increases is well accepted. Anderson (1980) states that "virtually every study of skill acquisition has found a straight-line function on a log-log plot," where the plot is of time to perform a task versus number of times performed or amount of practice. While he does not say the line slopes downward (decreased task time with increasing experience), all examples given show that behavior.

The same user who resisted learning a complex language may come to resent the simplified one he did learn because it proved too primitive for his later needs. This assumption is not well established. If true, this behavioral trait can be of importance in the design of the language for many data processing or database systems. An example in computer programming is found in the languages BASIC and APL. BASIC is easy to learn and use. Indeed, its

ease of use helped to popularize interactive computing. But, it is limited in logical power, a fact that may not be discovered until after the user has learned the language and come to understand what logical power is lacking. APL, on the other hand, is daunting to the novice programmer, but of impressive logical power once learned.

One important aspect of this study is the attempt to demonstrate that users most appreciate a language they can quickly learn to apply to simple problems and gradually learn to apply to more complex ones. Indeed, it is likely that users raise their sights and take on more complex problems as their skill with the language increases (though this latter phenomenon is not studied here).

1.1 Background

Carlisle (1974) defines three major aspects of the man-machine interface: user cognitive complexity, environment (including the problem to be solved), and language complexity. Language complexity tends to stand for logical power, as well. There is no virtue in complexity per se, but it is assumed that power and complexity are highly correlated and interrelated, with complexity a consequence of power. We will generally follow Carlisle's scheme, but with different terminology. Carlisle also first proposed that there exists a best match of interface complexity with a given combination of user and problem.

It is necessary, of course, to conduct a study of adaptation over time. In general, user behavior studies of query or programming languages have not emphasized this. Instead, they have tended to measure a user's performance at one particular point or over a very short interval. We feel it is essential to study how individual users learn (or fail to learn) to use a language over a period of time. Furthermore, we do not expect that all users should reach the ultimate level at the same time. Conducting a longitudinal study introduces a number of practical problems (see section 2.7.2). Even at the outset of this study, we felt a desire to conduct another over a much longer period of time.

A number of previous studies that bear on this one were discussed as background for this study in an earlier paper (Meadow 1981) and need not be repeated here. We restrict ourselves, instead, to the few studies that had the most impact on our own thinking.

Dolby (1971) discussed the problem facing a person considering what language to learn and use for documentation-related applications (generally information retrieval). He pointed out that a prospective user must make an investment in a language (presumably the time, energy, and expense of learning it and the cost of converting existing programs). Clearly, an investor expects a return on his investment, but, as a novice, he may be unable to assess the benefits of learning any given language; i.e., he may lack the knowledge and experience necessary to render such judgments.

In this work, we do not make a distinction between programming and query language. Both instruct a computer what functions to perform. Although there may be syntactic differences between the typical representatives of either class of language, we do not believe they differ in any fundamental way.

Mooers (1960) made a similar point in his Mooers' Law:

It is my considered opinion, from long experience, that our customers [users] will continue to be reluctant to use information systems--however well devised--so long as one feature of our present intellectual and engineering climate prevails. This feature...is that for many people it is more painful and troublesome to have information than not have it.... [We] can expect that they will tend to avoid using a retrieval system which is efficient at putting information into their hands. This contradictory principle can be stated:

MOOERS' LAW: An information retrieval system will tend not to be used whenever it is more painful and troublesome for a customer to have information than for him not to have it.

Mooers' emphasis is on the user's reluctance to undertake a task rather than to learn the language, but the effect is similar.

Benjamin Whorf (1956) provides a basis for understanding the language selection dilemma. The Whorf Hypothesis (which seems to have been formulated posthumously as an encapsulation of his work and was not something he put into words himself) states that the language one learns determines one's view of and approach to life, including problem solving. Whorf studied natural languages and their impact on user behavior, but the hypothesis carries over into the field of artificial languages, as well. The first natural language learned is, for most people, the only language learned, and linguistically related behavior must then be ascribed to it. But any user of an artificial language can always discuss his problem in natural language and describe (program) it in a programming one. Thus, his problem-solving behavior is not quite so strictly within the limits imposed by his programming language. Research is being conducted into the application of natural language to computers (Walker 1981); however, the acquisition of natural language does not prepare us for using it to instruct a computer.

1.2 A Conjecture on Language Learning

If we combine the Dolby, Mooers, and Whorf concepts, we may synthesize a conjecture based on them. The first-time user of an artificial language is faced with a cost-benefit analysis which his inexperience makes him ill-equipped to perform. His perspective is likely to be biased toward the first language he chooses to learn, and that choice is likely to be determined by ease-of-learning rather than power-in-use.

Those who first learn a language of low complexity (power) may come to perceive computer problem-solving ability as quite limited.

Even when no apparent choice is offered, a user normally has a choice of how extensively he is going to learn, or how intensively he will try. Hence, by different degrees of learning, users will, in effect, exercise a choice of the language they acquire, perhaps creating in their own minds a subset of the language being presented.

Thus, the conjecture is:

For a person using a computer system as a problem solver, the perception of both the problem set that is appropriate to attempt and the acceptability of performance and solutions is governed by the problem-solving language of the user. Minimal language knowledge leads to minimal perception of possible solutions, and minimal perception of possible solutions leads to minimizing language learning.

Given this conjecture, we may reexamine the results of several studies that seem to suggest that, at least in bibliographic searching, there are few differences in the language-use behavior of moderately experienced users and highly experienced users. Fenichel (1979) and Lodge (1980) found that, in terms of frequency of use of the various commands available and syntactic errors, there was essentially no difference between moderately experienced and highly experienced groups. Intuition would have suggested the contrary--that more experience would lead to more sophisticated, different usage. The lack of difference may be attributable to the previously stated conjecture. These searchers may have formed their outlook on what a bibliographic search should be by learning one of several similar languages. Their training may have provided them a limited view of what problems could be solved; hence, they did not try to solve more complex ones, even as they became more experienced. Clearly, this explanation is a conjecture and would have to be proven experimentally. It is not our claim to have proven the conjecture here, but to be moving toward a proof. We do claim that the way a problem-solving information system is presented colors the way a user will try to solve problems.

1.3 Objectives and Assumptions

Our purpose is twofold: (1) to investigate behavior as users begin to use artificial language and (2) to identify factors that affect their rate of adaptation (level of performance and attitude). A new user of any language goes through such an adaptation period. (We use the term adaptation rather than learning because, during this period, attitudes are being developed [Whorf] as learning takes place.)

The adaptation period may last from a few hours to a few years. While it has a fairly precise starting point (first training class, initial exposure to a users' manual, etc.), it has no well-defined end. It is analogous to a boundary layer in fluid dynamics--a transitional interface between fairly well-defined behaviors (no use and journeyman use). The beginning of the adaptation period is well-defined, but the end is not.

The study was devoted to measuring user performance and attitude during the adaptation period. Experimental subjects were selected at three levels of prior computer experience. A database was constructed with the intent to minimize user differences in subject matter knowledge. Three languages (query systems) were constructed to search the database. These languages varied in complexity and power. They were designed so that the second language contained all the features of the first plus new ones, and the third all those of the second plus new ones. Each user was taught only one of the search methods and asked to perform twenty searches. Various measures of performance and attitude were taken.

This study does not involve the statement and testing of formal hypotheses. We contend that, in this field of human-computer interaction, most experienced researchers acknowledge that: (1) we do not know exactly what variables to measure; (2) we often do not know how to take measurements; and (3) we often cannot accurately relate measurements to the more global behavior actually under study. Hence, nothing is lost at this stage of research in the field by concentrating on searching for trends and relationships, rather than testing hypotheses that we cannot prove actually govern the behavior of interest.

Some examples of measurement and application difficulty include the traditional (to bibliographic studies) measures recall and precision, on which we comment further in sections 2.8.3 and 3.1.2. These measures are often taken and cited in bibliographic studies, but are, we believe, of unproven relationship to searcher behavior. Another example is that composite of human behavior collectively called cognitive style or, roughly, problem-solving style. Most of us recognize this as an important factor in predicting human performance or learning, but we do not know how to measure it, so we tend to select surrogates. These may be experience, undergraduate major subject, or even age.

For many experiments concerned with how humans solve problems with the aid of a computer (retrieval of information, calculation of some quantity, editing of a text, etc.), the temptation is nearly irresistible to use a standard set of problems for all subjects. (We succumbed to this urge in this study.) However, we are unable to measure the motivation of the subjects in a test situation or the effect of motivation on behavior when the problem has no real-world meaning to the subject. In his admirable dissertation, Carlisle (1974) claims high motivation for his subjects, who were largely undergraduates playing a space-age war game. But, his attempt to measure cognitive style or complexity of the players went awry because of suspicion on the part of the tester that the subjects were not serious in taking the cognitive-style tests.

Therefore, our approach represents an attempt to draw a rough sketch rather than to paint a detailed picture. We have made some assumptions that are not formal hypotheses, and we have attempted to see patterns rather than to prove relationships definitively. We must use the same data to determine what relationships matter and to form conclusions from them. We believe the results to be no less valid than if more specific hypotheses had been posed and tested. This study is not an end. We hope it is a beginning.

In lieu of formal hypotheses, we undertook:

1. To investigate the behavior of users of an interactive database search system that is new to these users.
2. To determine whether there are differences in user performance or attitude toward the system as a function of the following factors:
 - a. prior experience (as a surrogate for cognitive style),
 - b. level of complexity of language used, or
 - c. experience with the use of the language.

3. To treat this as an exploratory study, with the following practical consequences:
 - a. As a first study of the adaptation process, the variables selected for study are not predetermined to be the correct ones. Which variables are of importance is something to be learned from the study.
 - b. The sample size available was smaller than desired.
 - c. Suggestions on future research methodology are part of the findings of the study. Specifically, we assumed the following:
 - (1) Performance will improve with experience.
 - (2) Attitude of users toward a language will, in general, increase as the language is being learned, then peak, and eventually decline as users master the language and discover its weaknesses.
 - (3) Users with more prior, related experience will be in a better position to evaluate a language, or they will at least express opinions about one.
 - (4) Performance of users will generally increase with increasing prior experience, experience with the new language, and language power (often expressed as complexity).
 - (5) However, the best match of a user with a language is not necessarily found by giving him the most powerful and complex language, but the most powerful one his skill level is able to handle.

1.4 Summary of Results

Results are given in chapters 3 (individual conclusions, often tentative, from statistical tables) and 4 (overall conclusions). In summary form, they are:

1. Performance does increase with experience.
2. Attitude changes minimally with time (experience with the language under study), but enough to suggest that this phenomenon may be seen more sharply if a study were conducted over a longer time period.
3. Differences in attitude as a function of prior experience and of language level used exist.
4. The best user performances were achieved by the least experienced group of users with the least complex language and the most experienced users with the most complex languages. Thus, the case for matching users with a language designed for their particular level of skill seems to be made.

2. DESIGN OF THE EXPERIMENT

Our intent was (1) to measure some aspects of behavior and attitude over time as the users performed a series of database searches and (2) to test the hypotheses that (a) performance of users is a function of prior experience and complexity of the language and (b) attitude toward the language, as a search tool, begins to decrease after the language has been learned (either proving unable to provide satisfactory solutions to new kinds of problems or causing the user to perceive the solutions as in some way lacking).

In brief, the experiment tested users having three levels of prior computer experience and three levels of language complexity/logical power. Each user was trained in one language and then performed twenty searches, in five sessions, scheduled one day or more apart. Several measures of performance were used (see section 2.8), and one attitude questionnaire was administered at the end of each session.

The remainder of this chapter is a description of the design of the experiment. Results and conclusions are treated in chapter 3.

2.1 The Database

The database consisted of some 580 records that were condensations of news articles appearing in British national newspapers between October 1980 and March 1981. The text was limited to 150 characters. An example of a data item is:

BRITISH STEEL CORPORATION CLOSED ITS
CONSETT WORKS ON 15 SEPTEMBER 1980.

The data items were classified by subject, using a hierarchical system of subject headings. The subject headings were also included in the database. The major subject heading divisions were:

General News

- U.K. Economy
- Middle East
- Celebrities

Financial News

- Stock Exchange Prices
- Foreign Currency Exchange
- Commodity Prices

Sports Results
Football
Racing
Golf

Records containing only subject headings are called index records. An example of an index record is:

CHOOSE FROM AMONG THE FOLLOWING:

1. IRAN
2. IRAQ
3. ISRAEL
4. RETURN TO ROOT *
5. QUIT *

* Not part of the index, but
offered as a choice to the user.

Each index record has one or more subordinate records which may be index or data records. In the example, there is an indication of three subordinates: IRAN, IRAQ, and ISRAEL, each of which is itself an index record (although this fact cannot be determined from this record). The user would have to follow the path and see the actual record to determine if it contained data or more pointers. Data records have no subordinates. The entire structure forms a hierarchy or tree.

The first index record is called the root. It is the record at the apex of the hierarchy--nothing points to it as a subordinate. Depending on the path followed, there might be five to eight levels of index structure.

A dictionary contains every word occurring in an index or data record except those words appearing on a stop list, a list of common words assumed to cover no information.

A more complete description of the database and its structure is given in Appendix A.

2.2 The LURES Search System

The search system employed in the study was called the Language Use Research System (LURES). LURES consisted of three search languages (with interpreters) for searching a single database. The language for any user was determined by a character in his password, and there was no switching among languages by users. While some languages included features of other languages, each was taught to users as an integrated whole, not a composite of other languages.

2.2.1 General Description

When starting a search, a user logged in by providing his password, the question number, and the search number. Each question was uniquely numbered, and the search number was a composite of session number and search within session.

Upon completion of log-in, the root index record (see section 2.1) was automatically displayed without further effort on the part of the searcher. Then, the search was conducted. When the searcher indicated he was finished, he was asked two questions concerning his evaluation of his own performance in respect to (a) the quality of the answer to the question and (b) how well he performed the search. These questions were asked through the computer and the answers entered directly into the computer. The answers were used in the computation of performance data. The text of the questions is given in Appendix B.

At the end of each session of four searches, a printed questionnaire was administered concerning the attitude of the searcher toward the search system, and this formed the basis of the attitude measurements. Note that the user's attitude toward his own performance was included among the performance variables.

The three search systems introduced progressive levels of logical power and complexity. Each language included all the logical capabilities of its predecessors. Language, or search system, 1 is a nonprocedural system, i.e., the user does not issue commands, but indicates choices among options offered to him. This is also known colloquially as a "menu" system, but was called a "choice-of-path" method in our training.

Once the root record was displayed, it offered a choice of any of the subordinate subject headings, and those offered further subordinates, and so on. A data record has no subordinates. In addition to these choices, there was always the option to return to the root, which enabled the searcher to start down another path of the hierarchical tree or to quit.

2.2.2 Language 1

The root is automatically displayed upon completion of log-in. At the end of the display of the record, the following options are displayed:

(Text of record)

1. Subordinate 1 (if record is an index record)
2. Subordinate 2 (if record is an index record)
- 3-5. (up to five subordinates)
- N-1. Return to root (unless the current record is the root)
- N. Quit

The searcher must decide at each index record, or node, which subordinate path to follow. He does this by typing in the number indicated. The maximum number of choices is seven--up to five subordinates, return to root, or quit. If the path proves to be the wrong one, the only option is to return to the root and come back down again on another path. This limitation was a conscious design decision. Other options might have been possible. We wanted the differences among the three levels of language to be almost even, fully granting the lack of a quantitative definition of "difference." This decision represented a consensus among several persons experienced in information retrieval.

2.2.3 Language 2

The second language is a procedural one, offering the user a choice of commands with which to search the database or to learn or relearn how to use the system.

The commands available (in addition to the L1 choices, always available for use) are listed below and are discussed more completely in Appendix A.

1. PAGE [UP, DOWN, LEFT, RIGHT]. (PAGE followed by one of the words in brackets.) Use of this command enables a user to retrieve the record immediately above (the parent of) the current record [UP]; the first subordinate of the current record [DOWN]; the next sibling [RIGHT] or the previous sibling [LEFT].
2. FIND <term>. The symbol <term> means to replace it with any single term, without brackets, as FIND BRITAIN. The command results in the creation of a set of records containing the term. Only single words are permitted. The term must be in the system dictionary if a non-null set is to be created. The display is always followed by a statement of the number of records in the set, including zero. The first record of a set is automatically displayed. There is at most one set in existence at any time. If a new set is created, the old set is erased. If a null set is created, it does not erase the existing set. Records are ordered by level number and record number. The level is the level within the hierarchy, numbering from 1 at the root record. (The record number is automatically assigned as the database is built and has no semantic significance.) The usual choices of subordinates, return to root, or quit, are offered whenever a record is displayed. The Language 2 user may then issue another command or indicate a Language 1 choice among the options offered.
3. PAGE SET [DOWN, UP]. This command enables the searcher to step through the records of a set, one at a time, for viewing. There is no significance to the ordering of records within a set. PAGE SET DOWN goes from the beginning toward the end of the list of records that comprise the set. PAGE SET UP does the reverse. Note that, unlike many systems, there is no PRINT command that will print multiple records with one command. PAGE SET cannot be used unless a set has been created.
4. DICT <term>. This command enables the user to search the dictionary. The term is found and a display is presented that consists of the search argument and the two words preceding and two following in alphabetical order. Displayed with each word is the number of records of the database in which it occurs. If the search term is not in the dictionary, it appears in the display with a note to this effect.

A typical result of a DICT command would be of this form:

:::DICT IRAN (User enters the command)

Term	No. Records
INTRUDE	10
IRA	1
IRAN	27
IRANI	6
IRANIAN	3

5. PAGE DICT [DOWN, UP]. This command enables the searcher to page or scroll through the dictionary, moving five words at a time, up or down in alphabetical order.
6. HELP. This command gives access to a file of instructional material (reproduced in Appendix A). This file is searched using the rules of Language 1, except that instead of an option to quit, the searcher is offered the option to return to the search.

Language 2 offers considerably more power than does Language 1, but users often quickly note the absence of any means of finding records containing a combination of two or more terms.

2.2.4 Language 3

The third language differs from the second primarily in the extension of the FIND command, which now permits searching on the basis of more than one term. The added commands are:

1. FIND <term> ":". The symbol : is a truncation symbol and means to find all records that contain any word beginning with the root <term>. Thus, FIND IRAN : retrieves both IRAN and IRANIAN.
2. FIND <term1> [AND, OR, NOT] <term 2>. This is a conventional Boolean search command. One of the options AND, OR, or NOT must be used. Each term must be a single word.
3. FIND /P/ [AND, OR, NOT] <term2>. This is a variation on the first form of the Boolean command. The symbol /P/ refers to the previous, or current set. The existing set is combined with a new one based on <term2> to form a new set. Clearly, a previous set must have been defined if this is to be used.
4. FIND [AND, OR] <term1> <term2> . . . <term5>. This permits sets to be defined using up to five terms, each of which is linked by AND or OR. At least two terms must be used. FIND AND A B is identical in result to FIND A AND B.
5. HIST. This enables a searcher to review the search by displaying the FIND (set creating) commands successfully executed during the current search.

2.3 Experimental Subjects

2.3.1 Recruiting

In recruiting subjects, our intent was to find three groups, varying in prior computer experience, and as homogeneous as possible in other respects likely to bear on the project. Recruiting inexperienced and highly experienced people proved no great difficulty. Unexpectedly, finding the middle group did.

Inexperienced subjects were recruited from among postgraduate students in education at the University of Sheffield and the Sheffield City Polytechnic. A short questionnaire was enclosed with a letter sent to all who volunteered, and on this they were to record their computer-related experience. An operational definition of "inexperienced" was that the subjects had never taken as much as a term's formal study in computer programming or bibliographic searching.

Experienced subjects were recruited primarily from among the staff of the University Computing Services Department and third-year students (the equivalent of American seniors) in computer science at the Polytechnic. An operational definition of 'experienced' was that they have at least one year of working experience in the field and either a second year of experience or a year of academic training.

Moderately experienced subjects were operationally defined as falling between the other two groups. Ideally, we would have liked people with exactly one academic term of training. However, university schedules made the most likely group unavailable during the period when experiments were to be run, and as the principal investigator had a fixed period of residence at the university, this scheduling was critical. The majority of middle-experience subjects were recruited from among postgraduate education students who had had some computer experience.

A total of 50 subjects were recruited. They were selected with the goal of having an equal number in each prior-experience category and were then to be randomly assigned to a language-level group with the intent as having those groups of equal size also. Ideally, we would have had a 3 x 3 array with five persons in each experience-language cell. Because of last minute dropouts, we used a total of 44, with the following actual experience-language array:

	5	5	4
L	4	5	5
	6	5	5
E			

Number of Subjects by Prior Experience (E) and Language (L) Used

The table below shows the distribution of education and experience among the volunteers. Four questions were asked on the questionnaire:

Table 1. Experience Characteristics of the Sample

Questions	Experience Group		
	E1	E2	E3
1. Ever written a computer program or online bibliographic query?	13 No 3 Yes	2 No 13 Yes	1 No 14 Yes
2. Had formal training in computer use?	15 No 1 *	1 No 14 Yes	1 No 13 Yes
3. Had a full term's academic training in computing?	16 No 0 Yes	8 No 7 Yes	2 No 12 Yes
4. Had work experience as a computer programmer, systems analyst, or biblio. searcher?	16 No 0 Yes	13 No 2 Yes	0 No 14 Yes

* One subject reported having had one lecture.

2.3.2 Training

The participants were randomly distributed into separate training sessions for each language. Subjects were offered a choice of times and dates for the first instructional meeting, and the entire group coming at any one time was assigned the same language. Eleven sessions were held, with the number of participants at each ranging from one to seven.

As each training session might have had a mix of all experience levels, a fixed format was followed for each language session to insure that the presentations were the same for all groups. Languages L2 and L3 were taught as extensions of L1 and L2, respectively, so that each session began with an explanation of the database and language L1. Then, as much additional material was introduced as was appropriate for the language being taught.

Instructional materials consisted simply of a set of charts illustrating salient points, and a handout (Appendix C) summarizing the lecture content. The handouts, like the sessions, were designed in increments. All got the L1 description. Then, L2 users got two additional pages describing L2, and L3 users got yet another page describing L3. The terminals available were not of the dial-up type, and there was no facility available in which we could both teach and offer a demonstration to a group even as large as seven. Hence, no demonstration was included in the training session. To partially make up for the lack of user trials before the individual began his experimental work, the set of questions used in the first session was intended to be easier than succeeding ones (see section 2.4).

Subjects were advised to bring their handouts with them to search sessions, and it was suggested that bringing paper and pencil would be useful. They were told that a project staff member would be present during all search sessions.

2.4 Questions

Twenty questions were developed for the experiment. These questions were classified by expected level of difficulty. The classification was based on the language for which the question was written, with the proviso that all questions should be able to be answered in any language. Definitions of the categories were:

Level 1 questions were so written as to provide clues in their wording to the decisions to be made by the searcher as he used choice-of-path searching. In most cases, use of the FIND command would have speeded and simplified the search. All Level 1 questions called for only a single record as the answer, although this was not explicitly told to the subjects.

Level 2 questions contained sufficient ambiguity in their wording to encourage the use of DICT and FIND commands, and these, in turn, usually required use of PAGE commands as well. More terms may have been suggested than may have been actually needed to complete the search, but the wording would not necessarily have guided the selections in choice-of-path mode. (The actual words of the question may not have been the same as the words of the subject headings in the index.) More than one record might have been required in the answer.

Level 3 questions make the use of FIND a near requirement, and more than one data record had to be examined to determine the answer.

Each session included a mix of questions at different levels. A table in section 3.1.4 shows the number of each type of question used in each session. From this, a level of difficulty index was computed for each search session (of four searches).

Questions were worded as if they were being assigned to a research assistant, with the intent that the identification of records, not an extract from their contents, constituted the answer. In other words, the answer to a question about the price of sugar on the commodity market was not a price, but the number of a record containing the price. All participants searched the same questions in the same sequence. They were given only one question at a time, each written on a separate sheet of paper which contained, in addition to the question text, the user's password, search and session number, and question number. Searchers were asked to record the numbers of the records that included their answers to any question.

2.5 Scheduling

Ideally, we would have liked to schedule all search sessions with uniform intervals between them, and two days was arbitrarily selected as the interval. The point was to give the instruction and experience time to sink in. However, subjects were generally unable or unwilling to adhere to such a rigid schedule. We compromised on trying to maintain a one-day separation, but we still had to accept considerable variation.

2.6 Pilot Project

In March 1981, just before a five-week university holiday, we conducted a pilot study, with the primary objective of debugging our procedures. The subjects were twelve students in the one-year master's degree program of the PGSLIS, and all were selected to fit the definition of moderately experienced users.

Subjects were trained in the manner intended to be used in the main project. They were then asked to do twenty searches in four sessions of five searches each. (The main experiment used five sessions of four searches.) A brief questionnaire used at the end of each session asked for:

1. Their opinion of the training session--whether it was adequate to prepare them for searching.
2. Their opinion of the search system as a whole, expressed on a scale of 1 to 5.
3. Their comments on the search system.

A large number of search system software errors were uncovered. Fortunately, the five-week holiday period gave us time to correct most of them.

A number of comments on training were received and used to modify the presentation. The most persistent comments had to do with training in the mechanics of using the terminal.

The single evaluation question showed very little variation in answers, but the free-form responses to the question asking for comments, when categorized, began to show a pattern indicating that subjects were separating system elements in their own minds, i.e., not treating the system as a monolithic whole. As anticipated, these views showed signs of change over time. As a result, we decided to change the attitude questionnaire into one separately asking about major search language features, database elements, and an overall question about the entire system.

We also used the pilot to help debug the questions. The main problem with them proved to be inadvertent Americanisms. For example, one question, in an attempt at lighthearted wording, included the expression "carriage trade," meaning "social elite." This expression was unknown to many of the subjects. Several assumed it to have a transportation connotation, misleading in the context used. We also found the verb "gotten" to be distracting, since it does not occur in British English.

We felt there had been enough rough spots in this experiment that we did not perform a detailed statistical analysis of the records of the searches.

Our conclusion about the pilot was that it proved an invaluable aid to setting up the main experiment. In fact, we would have been better off to have repeated it entirely, once changes in design were made. The shortage of subjects prevented this.

2.7 Conduct of the Experiment

In this section we describe the arrangements for carrying out the experiments.

2.7.1 Arrangements for Searching

1. Facilities. The computer system used permitted only one user at a time to access a file. Hence, it was necessary to make a complete copy of the system--programs and database--for each of the persons who used it simultaneously. The university maintained a central terminal users' room and also supported a few terminals located in departmental offices. In order to insure that users could find a terminal free at an agreed time and that we could accommodate up to three people simultaneously (a scheduling necessity if we were to finish on time), we requested and were granted priority use of three terminals in a relatively secluded part of the Computing Centre. This enabled us to avoid distraction for our users and to avoid scheduling difficulty as a factor in performance and attitude.

As with most time-sharing computers, system response time varied with the number of simultaneous users. The number of computer users not related to our experiment was not under our control, but, based on informal observation, we do not believe response time varied enough to have a significant effect on users. No subject volunteered a comment on reaction time. In general, we were satisfied that the conditions under which users worked were realistic (similar to conditions for those who perform this function professionally) and uniform for all subjects.

2. Scheduling. We would have liked to schedule all training and search sessions at uniform intervals--one or two days apart. We simply could not get all subjects to adhere to such a regimen. Instead, we tried to maintain a minimum of one day between any two sessions and to hold the maximum to one or two days. We were not very successful in imposing a maximum interval; however, we had only a limited violation of our minimum, caused by subjects who were planning to be away for extended periods. The table below shows the distribution of intervals between sessions expressed in hours.

Mean	50.41 hours
Median	28.25
Mode	23.00
Minimum	6.00
Maximum	338.00

n = 176 intervals between sessions

3. Search sessions. A project member, whom we shall henceforth designate the attendant, was present during all search sessions. This person's duties were to:
 - a. Log users into the computer. (Logging in and logging out were not considered part of the search; hence, we were willing to perform these functions for the subjects.)
 - b. Give out the questions. Each question was printed on a separate piece of paper which contained the question number, search number, and question text. It also had space to record the record numbers which constituted the answers to the question. These were used later in computation of precision and recall. The user's password was added to the paper by the attendant.
 - c. Record in a log the number of each question as it was given to the searcher, so that the log represented a continuing record of what was occurring, rather than what was planned. This was done to insure that, in case of a problem (such as a computer failure), we knew accurately what was supposed to have been recorded in the computer file. The attendant was also to note any unusual occurrences during the search, such as a software error or accidental striking of the BREAK key, which aborted the search.
 - d. Make appointments for future sessions and change the appointment schedule as necessary.
 - e. Assist the subjects in case of any mechanical problems with terminals or difficulty in understanding the meaning of any system message sent by the computer.
 - f. Not to provide any substantive assistance or advice on how to search or answer the question or interpret the meaning of database items.
4. Data collection. The majority of data descriptive of a search was automatically recorded during the search. A few other items were acquired by computer-generated questions before and after the search, and still others were recorded on paper for later entry into the computer. The use of paper forms created some record-keeping problems and necessitated special programs for entry into the computer, but there was not time beforehand to prepare the software needed to avoid use of paper. Data elements collected, and the means of collection, are listed below. Some detail of physical record structure is included as an aid to anyone who might work with this data in the future.

- a. Prior to the search, immediately following login, the following elements of information were requested of the searcher by the computer:
- (1) User password. A 4-digit code uniquely identifying the subject in all machine-stored records. Digit 1 encoded experience level (1 = no prior experience, 2 = some prior experience, 3 = much prior experience). Digit 2 encoded language assigned: 1, 2, or 3. Digits 3 and 4 constituted a randomly assigned serial number for the searcher. During the experiment, 48 persons were registered as subjects; 45 persons actually participated. The project staff used serial numbers 00 and 99 to indicate searches done by them for training or debugging.
 - (2) Question number. A 7-digit code uniquely identifying the question (as distinct from the search). Digits 1-3 were random numbers (to make the entire number appear random in order to conceal from users the question difficulty classification), digit 4 was a difficulty code (see section 2.4), and digits 5-7 were a question serial number. Some 30 questions were prepared, but only 20 were used. (These questions were administered to all participants.)
 - (3) Search number. A 2-digit number, the first digit of which encoded the session number (1 to 5) and the second the search within the session (1 to 4).
- b. During the search, the following data elements were recorded in the computer-stored record of the search:
- (1) A heading on the record of each search, consisting of:
Date of search (Note: Because all searches but one were performed during May, 1981, we modified the date on the one June search to the fictitious May 32, 1981, to simplify date calculations.)
User password
Time search began
Question number
Session/search number
 - (2) For each user command or selection of a choice, the following data elements were recorded:
Time
Sequence number of command or choice (1,2,3,...)
Text of command or choice
Command type code
Error status code
Record pointer (This indicated the record number in the database retrieved by the command or option.)

(3) At the end of the search (cued by the command "QUIT" or a quit option), the searcher was asked the two search evaluation questions (see Appendix D). Answers were integers from 1 to 5. To maintain the format of a six-element record, the record was filled out with *****. There were two such records.

(4) As an aid to editing the recorded data, a tag line, consisting of 18 "(" characters, was recorded to mark the end of a search entry.

At the conclusion of each session (four searches), a questionnaire was administered to be completed and returned before the searcher departed (see Appendix E).

5. Logging. A log was maintained, containing the data elements listed below. This was recorded manually by the attendant.

Date

Time of day

User password

Session number

Question number (These were to be recorded as the person began work on a question.)

Questionnaire status (A mark was made when the session questionnaire was completed and returned to the attendant.)

Comments

A separate appointment log was kept, showing what people (by user password) were expected at what time of what day. A separate log showing the record of completion of searches by user password should have been maintained. Without it, checking on participants who were not scheduled for a next session was difficult.

6. Attendants. Quite a considerable amount of knowledge and tact was required of the attendants who worked directly with the users. Their task was simple when everything ran well--to start up the computer, maintain the logs, feed questions, and collect the questionnaires. It was also required that they not say anything to one searcher that might give another searcher a clue on how to search or to draw attention to the fact that different languages were being tested.

When things did not go well, usually as a result of software problems, the attendant was required to make a decision on the spot. Because the attendants did not know the software well, this became the greatest source of apprehension during the running of the experiment. But no irreversible errors were known to have been caused by these problems. Four persons shared the duties of attendant: the project director, the database designer-trainer, and two persons recruited

especially for this task (hence, having the least knowledge of the software). The latter two should have been recruited much earlier in the project in order for them to come to understand the software better.

2.7.2 Problems

There were a few recurring, unanticipated problems the effect of which on results we cannot estimate. These are discussed here primarily as a warning to others who may plan similar studies.

1. Software. We have noted several times that some software errors did occur during the running of the project. Since special-purpose software was required, in retrospect there seems little that could have been done about this. Tests were run by the programmers and by several members of the staff of the PGSLIS, but 45 people can catch more errors than 5.

The most serious of these was too deep a nesting of subroutines. The programming language used was BASIC Virtual Memory on the Prime 750 computer. A maximum depth of nesting of 16 is permitted, and whenever this was exceeded it was a programming error, not an intentional usage. When the condition occurred, it caused the program, and hence the search, to abort. This was not the fault of any single program error, and, after the first two days, most of the causes were found and eliminated. However, the problem recurred occasionally throughout the experiment. We eventually realized that searchers could simply log back on, using the same search number, and continue the search. This created a few erroneous entries in our computer-stored record of the search, but as soon as these were detected, they were individually deleted by use of an editing program.

Another error was an intermittent failure of PAGE LEFT and PAGE RIGHT to recover the correct record. Repetition of the command usually got the desired result. Most users quickly discovered this and were not troubled by it. We found it a curiosity that even experienced programmers seemed cheerful to accept that sometimes a command worked and sometimes it did not. A more likely response, we thought, would have been for experienced users to be bothered by intermittent errors as unmachine-like, while the less experienced might not have thought computer inconsistency to be unusual.

2. Hardware. The computer we used (Prime 750) encountered a major fault just as the experiment was beginning. The computer was taken out of service twice during the first two days of the experiments and once later. Fortunately, the amount of downtime was little, overall. Some rescheduling of participants was necessary, but they accepted this with good grace.

A potentially more serious problem was in the design of the terminal keyboard. The ALPHA LOCK key made all alphabetic characters uppercase without use of the shift key. This was the preferred mode when using LURES. Unfortunately, the key was just above the shift key and to the

left of the A, a position in which it could easily be struck by accident. This happened at least once (so noted in the log--it may have occurred more often and not been noted) and confused the inexperienced searcher, since & a is not the same character as & A, and a command such as "page up" instead of "PAGE UP" would be rejected by the search system.

A similar problem occurred with the BREAK key, which has the effect of aborting the running program. This key is adjacent to the RETURN key, which must be struck for each command or choice. Hitting the BREAK key accidentally occurred at least once, and possibly more times. In this case, the searcher simply began again, as was done with the subroutine nesting software error, and we were able to correct the problem by editing the search record. Some amount of time was wasted, however, and some attitudinal effect may have occurred.

3. Subjects. Prior to the start of the project and the arrival of the principal investigator in the United Kingdom, it was assumed that subjects would be easy to recruit and that sufficient numbers for the middle experienced group and some for the very experienced group would be found within the Postgraduate School of Librarianship and Information Science. These groups were to be augmented by computer science students and some (at that time unspecified) inexperienced students.

We quickly found that computer science was a new discipline at the university. There were few students enrolled, and these had not had much experience. We also found that the combination of the scheduling of the spring holiday and examinations for the PGSLIS students prohibited their participation until the beginning of June, too late to assure the experiment's completion. Hence, we found it necessary to look elsewhere for subjects.

Since this project was designed overseas before the principal investigator's arrival in the United Kingdom, without the availability of detailed scheduling information, little could be done other than to accept the limitations and work around them.

Another subject-related problem was individual scheduling. As noted earlier, we had originally hoped that sessions would be a uniform interval apart, but found that users could not or would not accommodate that degree of rigidity in scheduling. Several times, subjects simply quit in mid-session. We had no choice but to ask them to finish the session as soon as possible. There were too few subjects to discard these persons' data. The dilemma experimenters face in this situation is whether to accept these variations, which almost certainly add to the variability of the data, or to limit the experiment. Since a longitudinal study was important to our objective, we felt it best to accept the variability with consequent degradation of the accuracy of the results.

While all subjects were asked not to discuss the project among themselves, unsolicited comments to the attendants made it clear that there was some discussion among subjects. However, we knew neither the extent nor the impact of these discussions.

2.8 Measures

In this author's opinion, research on information system user behavior has been characterized by the use of too many ill-defined variables of uncertain relationship to each other. This is caused both by the difficulty of measurement, in general, and by the lack of firm proof of the value of those measurements. In this study, we planned to use a relatively small number of measures: subject's background, environmental complexity, attitude, process performance, and outcome. The last two are composites, as indeed the rest may prove to be on further investigation.

In this section we define and discuss the measures taken during the study. All performance and opinion variables in this study are normalized to range between 0 and 1, with lower values indicating a negative attitude or poor performance.

2.8.1 Subject Background and Environment (Independent Variables)

We have previously discussed four variables in this category:

1. Prior experience, PX. Possible values are:
 - a. No prior computer experience on the part of the subject.
 - b. Moderate experience, e.g., some online search or computer programming, or some instruction in these subjects.
 - c. A minimum of one year of working experience in the field of computer programming, systems analysis, consulting, or online bibliographic searching, plus an additional year's work experience or a year's academic training in one of these fields.

By careful selection of test subjects, it appeared possible to force clean separation among these three groups, i.e., no ambiguity concerning to which category a given subject belonged. Also, we assumed we could control, or hold uniform, the amount of prior experience by selecting subjects from among students at the same level in computer-related studies (or having none). PX is an ordinal variable.

2. Interface complexity, IC. This is equated here with the language used. Possible values are:
 - a. Interface language L1
 - b. Interface language L2 .
 - c. Interface language L3

This, too, is an ordinal variable, there being no accepted measure of complexity of a language. Number of choices could be an approximation, but intuition suggests that this might have to be modified by probability of use of commands, with provision made for

any similarity or redundancy of commands. That is, one must consider whether the choice of next command, at any point, is equally probable or is a Markov function, heavily dependent on the immediately preceding commands.

3. Interface system experience, IX. This takes integral values between 1 and 5 and represents the number of search sessions that an experimental subject has undertaken. A much larger number, perhaps as high as 100, and ranging in time over as long as a year, might be desirable, but this is an economic impracticality at this time. IX is an interval variable, and it should be noted that it represents the number of times a user has worked with the system, not what was learned previous to the experiment. To this extent, it is an imperfect measure, but it seems superior to number of searches because of our intent to require a minimum of one day between search sessions. During this interval, some thinking about the system is bound to have occurred, and some maturation of understanding is likely over what might happen if all searches were done in a single sitting.
4. Question difficulty, QD. Although questions were planned so that they could be answered using any of the interface languages, they were designed to fall into the following categories, which then became measures of QD (see section 2.4):
 - a. Able to be answered using L1.
 - b. Possible in L1, significantly easier to accomplish in L2, and relatively easy in L3.
 - c. Possible in L2 and significantly easier to accomplish in L3.

There is, in effect, a level of question for each language. Ideally, these categories of question difficulty should have been verified by preliminary testing, i.e., the levels of difficulty should be verified by user performance, rather than asserted by their authors. Again, the matter of time and cost to do this prevented it from being done.

2.8.2 Individual Performance Measures--Process (Dependent Variables)

Following the approach of Carlisle (1974), we used two forms of performance measure: one for process, i.e., how the user goes about the search; and one for outcome, i.e., what results are obtained, regardless of how obtained.

Process variables: There is no established way to measure how well a search is being conducted, although a number of different approaches have been taken (Hewett 1981; Fenichel 1979; Marcus and Reintjes 1981a, 1981b; Penniman 1975; Wanger 1980). Our approach to process measurement was in terms of five variables: errors made, number of commands used, self-assessment of procedure, precision, and number of advanced language commands (in L2 or L3) used.

1. Freedom from error, ER, expressed as 1 minus the ratio of errors made to commands used. These are syntactic errors which, by definition, are errors in the statement of a command that render it impossible to be carried out by the language processor. We do not include any sort of strategic or tactical errors that reflect the way in which legal commands are used to achieve an objective.
2. Percent of maximum commands, CD, in any one search, expressed as 1 minus the ratio of commands used in a given search to the maximum number used in the longest search of the experiment. An alternative, which was expected to be highly correlated, would be consumed by time in the search. CD is a ratio variable.
3. Percent of advanced commands, PA, expressed as a ratio of the number of L2 or L3 commands to total number of commands or choices in the search. For L1 users, this value will always be 0, and it will not be counted in computing a composite performance score.
4. Self-evaluation of process, SP. This variable is the scaled answer to the question (see section 2.4) asking the user to evaluate his own performance. The scale ranged from 1 to 5 (5 = well done), and answers were divided by 5 to give a normalized range of 0.2 to 1.0.
5. Precision, PR. This is the ratio of the number of "correct" records retrieved to total number of data records retrieved during a search. The list of "correct" records was determined by the composer of the test questions.

2.8.3 Individual Performance Measures--Outcome (Dependent Variables)

In this experiment, questions were composed in such a way that the composer knew what records constituted "the answer."

1. Recall, RE, and recovery, RV. In bibliographic retrieval systems, we often use the measures recall (RE) and precision (PR) (King 1971), both based on the subjective concept of the relevance of a record to the question asked. Since we are assuming, in this study, that the relevance of a record is known to a high probability, while acknowledging that there may be differences of opinion which make this probability less than unity, we can use these measures individually and combine them to give a composite figure of retrieval merit based upon both recall and precision, called recovery, (RV). The rationale is that the usual choice facing a searcher is to achieve either high recall or precision at the expense of the other. To be sure of retrieving nearly all relevant records from a file, the searcher must usually retrieve many that are irrelevant, hence reducing the precision ratio. To insure high precision, he must usually limit his output to less than the full complement of relevant records. Neither one of these variables, then, is a complete measure of performance.

Further, there is no way of knowing which measure the user intended to optimize. The mean of the two gives a measure that is a function of both, and such a high value means a high value of both constituent variables. This we take to be a truer picture of a typical searcher's objective. Thus, we tested the variable

$$RV = (PR + RE)/2$$

where PR is precision and RE is recall (number of relevant records retrieved divided by number of relevant records in the file). RV ranges from 0 to 1 (perfect performance) and is a ratio variable.

2. Self-evaluation of outcome, SO. The user was asked to evaluate his outcome (see section 2.4) on a scale of 1 to 5, and, as with the self-evaluation of process, this answer was divided by 5 to achieve a range of 0.2 to 1.0.

2.8.4 Attitude Toward the System

After each session of four questions, subjects were asked to evaluate the system rather than their use of it. Questions were asked to elicit their opinions of the commands or choices, the database structure (including dictionary), and the system as a whole (see Appendix E). Questions were asked in three categories: those concerning system commands or choices; those concerning the database and its structure; and those concerning the system as a whole.

1. AC is the means of all command-related questions, normalized to the range 0.014 to 1.000. A zero value implied no response.
2. AD is the mean of all database-related questions, similarly normalized.
3. AS is the value of the answer to the single overall system, similarly normalized.
4. AT is the equally weighted mean of these other three means.

In the computation of these means, n, the number of individual answers, is the number of nonmissing answers, i.e., a missing answer, while entered into the computer as zero, was not counted in computing a mean. Thus, if answers to two database questions were 5 and 0 (0 denoting no response), the mean of the database questions, prior to normalization, was computed as 5.

2.8.5 Composite Measures

The composite measures of process performance, outcome performance, and attitude are:

1. Composite process measure, CP. This is the average of the independent measures of process. Its range is 0 to 1.
2. Composite outcome measure, CO. This is the average of the outcome measures, with a range of 0 to 1.
3. Composite performance measure, PERF. This is the mean of the two composite measures of process and outcome, also with a range 0 to 1.
4. Composite attitude measure, AT. This is a composite of the several attitude measures, again ranging from 0 to 1.

3. RESULTS

We begin this chapter with a brief survey of individual variables, their frequency distributions, and some measures of correlation among them. This is followed by descriptions of some important interrelationships among the variables, particularly of various measures of performance and attitude with prior experience (PX), interface complexity (IC), and time spent with a particular language, or interface system experience, (IX). We then consider the formal hypotheses of the study. All statistical results were derived using SPSS and the definitions of terms associated with that system (Nie 1975).

Throughout the chapter, some of the tables shown contain results that are inconclusive. We have not deleted these, but have systematically listed all tables generated in following our original plan. Tentative conclusions are presented following each major section, and final conclusions are presented in chapter 4.

3.1 Frequency Distributions and Correlations Among Basic Variables

3.1.1 Variables Related to Process Performance

1. Number of commands per search, CD, expressed as 1 minus the ratio of number of commands in a search to maximum number of commands occurring in the sample, or

$$1 - (\text{no. cmds in this search})/(\text{max no. cmds by any searcher})$$

Table 2. Frequency Distribution for CD

CD	f
0.0	1
0.1	1
0.2	4
0.3	0
0.4	3
0.5	16
0.6	21
0.7	91
0.8	215
0.9	527
1.0	0

n = 879

Table 2--Continued

Mean	0.83	Minimum	0.00
Mode	0.90	Maximum	0.90
Median	0.87	Skewness	-2.76
Std Dev	0.11	Kurtosis	11.55

The actual number of commands used tended to cluster near the low end of the scale, around 40. The largest number used in any search was 240, and all computations are based on number of searches as a fraction of this maximum. Since number of commands and elapsed time for a search seem closely enough related to bring into question whether or not they are measuring the same phenomenon, we correlated the two with the result:

$$r = 0.789, \text{ significance} = 0.000$$

Significance, in SPSS terms, is the probability of achieving the same result by random choice. A value of 0.000 means a probability less than 0.001.

It seems reasonable to conclude that number of commands and elapsed time do indeed represent essentially the same information; hence, we will use only one of them in future calculations leading to a measure of performance. We selected number of commands because the time measure includes delays imposed by hardware and software difficulties that presumably would not have affected number of commands as much as time.

2. Number of errors, ER, expressed as 1 minus the ratio of number of errors to total number of commands in a search (hence, a low value means high incidence of error), or

$$1 - (\text{No. errors in search})/(\text{no. cmds in search})$$

Table 3. Frequency Distribution for ER

ER	f
0.0	0
0.1	0
0.2	1
0.3	1
0.4	4
0.5	3
0.6	17
0.7	58
0.8	197
0.9	598
1.0	0

$$n = 879$$

Table 3--Continued

Mean	0.85	Minimum	0.20
Mode	0.90	Maximum	0.90
Median	0.88	Skewness	-2.68
Std Dev	0.08	Kurtosis	10.56

This is a highly peaked distribution, with over 90% of cases at 0.8 or higher (less than 20% of commands in error). Here again, we investigated whether ER was closely related to number of commands, with the result

$$r = 0.9992, \text{ significance} = 0.001$$

This was an unexpectedly strong relationship suggesting that the added practice in a long search does not reduce the likelihood of error, nor does the frustration of a long search increase it. They may perhaps offset each other. Whatever the cause, error rate maintains a close linear relationship to number of commands entered. Thus, use of both CD and ER in computing performance seems unnecessary.

3. Percent of advanced commands, PA, expressed as a ratio of number of L2 or L3 commands in a search to total number of commands in a search, or

$$(\text{no. L2 or L3 cmds in search}) / (\text{total no. cmds in search})$$

It should be recalled that "command" includes a choice exercised in L1, and that PA is necessarily zero (hence treated as "missing") for those subjects assigned to be L1 users, since they did not have the option of using L2 or L3.

Table 4. Frequency Distribution for PA

PA	f
0.0	94
0.1	85
0.2	79
0.3	79
0.4	72
0.5	88
0.6	39
0.7	26
0.8	27
0.9	4
1.0	7

n = 600
279 cases missing (L1 users)

Table 4--Continued

Mean	0.32	Minimum	0.00
Mode	0.00	Maximum	1.00
Median	0.30	Skewness	0.48
Std Dev	0.25	Kurtosis	-0.48

Here we have, for our data, a rare case of a flat distribution, the value of PA being evenly distributed over the range 0.0 to 0.5.

In correlating PA with CD and ER, the following results were obtained:

PA-CD: $r = 0.037$, significance = 0.138

PA-ER: $r = 0.046$, significance = 0.089

These coefficients are low enough to suggest that PA and CD or ER are measuring different phenomena.

4. Self-evaluation of performance, SP.

Table 5. Frequency Distribution for SP

SP	f
0.0	0
0.1	0
0.2	25
0.3	0
0.4	70
0.5	0
0.6	258
0.7	0
0.8	305
0.9	0
1.0	221

$n = 879$

Mean	0.74	Minimum	0.20
Mode	0.80	Maximum	1.00
Median	0.77	Skewness	-0.50
Std Dev	0.20	Kurtosis	-0.21

Correlations with other variables:

SP-CD: $r = 0.440$, significance = 0.000

SP-ER: $r = 0.442$, significance = 0.000

SP-PA: $r = 0.044$, significance = 0.098

Perhaps the only point worthy of note here is that use of advanced commands does not seem to bring with it any feeling of satisfaction for having done so, but use of a relatively large number of commands and a relatively low error rate are indicators of high self-esteem.

5. Precision, PR. This is fairly peaked, with 76% of occurrences between 0.0 and 0.2. No values above 0.6 occurred. One must bear in mind that there is no evidence that users are in fact trying to maximize PR, but intuition suggests that most people would aim in that direction. However, in an iterative search, which most are, once the "answer" is seen there is no incentive to make one more iteration in order to remove unwanted records from the answer set unless the search results are to be presented to someone else. In our case, it would not have been possible to increase precision artificially in this way, because PR was measured over all records seen, not just those seen after the last command. Hence, our PR values are less than those conventionally reported.

Table 6. Frequency Distribution for PR

PR	f
0.0	291
0.1	376
0.2	75
0.3	75
0.4	9
0.5	52
0.6	1

n = 879

Mean	0.12	Minimum	0.00
Mode	0.10	Maximum	0.60
Median	0.09	Skewness	1.50
Std Dev	0.13	Kurtosis	1.79

6. Composite measure of performance by process, CP. In keeping with our intent to normalize all measures to the range 0 (poor or low) to 1 (high or good), we can define CP as:

$$CP1 = (CD + ER + PA + SP + PR)/5$$

or, eliminating the redundancy between ER (number of errors) and CD (number of commands), as:

$$CP2 = (CD + PA + SP + PR)/4$$

Since PA is always zero for L1 subjects, the denominator should reflect that this variable is not always present. Hence, the denominators were changed to give:

$$CP1 = (CD + ER + PA + SP + PR)/D1$$

or

$$CP2 = (CD + PA + SP + PR)/D2$$

where D1 = 5 if an L2 or L3 subject, 4 if L1; and D2 = 4 if an L2 or L3 subject, 3 if L1.

As may be expected, these two versions of CP are closely correlated:

$$r = 0.929, \text{significance} = 0.001$$

Henceforth, we shall use CP2, simplifying the name to CP. Its frequency distribution figures follow. In all computations for CP, PCTADV is ignored for L1 users, i.e., the computation does not make use of PCTADV (necessarily zero for these users), but does not treat the entire case as missing. Only CD, SLFPROC, and PRECIS are used in these cases to compute CP.

Table 7. Frequency Distribution for CP

CP	f
0.0	0
0.1	3
0.2	8
0.3	59
0.4	249
0.5	308
0.6	189
0.7	56
0.8	7
0.9	0
1.0	0

n = 879

Mean	0.49	Minimum	0.10
Mode	0.50	Maximum	0.80
Median	0.49	Skewness	-0.01
Std Dev	0.11	Kurtosis	0.22

The result is a symmetric, bell-shaped curve with measures of central tendency all in the center of the range.

3.1.2 Variables Related to Outcome

1. Recall, RE.

Table 8. Frequency Distribution for RE

RE	f
0.0	52
0.1	2
0.2	0
0.3	11
0.4	0
0.5	43
0.6	8
0.7	31
0.8	0
0.9	0
1.0	732

n = 879

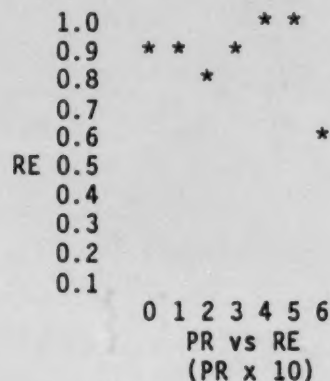
Mean	0.89	Minimum	0.00
Mode	1.00	Maximum	1.00
Median	0.99	Skewness	-2.18
Std Dev	0.27	Kurtosis	4.93

Eighty-three percent of users achieved a value of 1.0 for recall, i.e., few missed retrieving the complete answer. But, the largest group of those who did miss, scored zero. These probably include the ones who quit the search before completing it (see section 3.1.5). This distribution clearly does not present much variation and suggests that this measure is not too useful for "fact" type databases where questions are fairly well-defined with respect to the database.

2. Recall-precision curve. In bibliographic searching, it is generally accepted that RE and PR bear an inverse relationship to each other, approximately

$$PR = 1/RE$$

This is clearly not the case here, where the plot of PR versus RE is shown in the following figure.



Mean value of RE for all values of PR in interval 0-0.1, 0.1-0.2, etc. The curve is not of the usual shape.

3. Recovery, RV. Distribution data are:

Table 9. Frequency Distribution for RV

RV	f
0.0	51
0.1	7
0.2	47
0.3	17
0.4	21
0.5	616
0.6	72
0.7	48

n = 879

Mean	0.47	Minimum	0.00
Mode	0.50	Maximum	0.70
Median	0.50	Skewness	-1.78
Std Dev	0.15	Kurtosis	3.11

This is much more of a bell-shaped curve than that for RE or PR.

Correlating self-evaluation with recovery gives the following:

SP-RV: $r = .300$, significance = .000
RE-RV: $r = .913$, significance = .001
PR-RV: $r = .447$, significance = .001

RV, alone, is not a good predictor of the user's self-evaluation, nor vice-versa. The high RE-RV correlation is due to the high values of RE, compared with PR, that comprise RV. The fact that RV is so heavily dominated by RE in our data indicates that RV was not as useful a measure as initially assumed. Hence, we reverted to separate use of PR and RE in our various calculations of performance.

4. Self-evaluation of outcome, S0.

Table 10. Frequency Distribution for S0

<u>S0</u>	<u>f</u>
0.0	1
0.1	0
0.2	30
0.3	0
0.4	61
0.5	0
0.6	128
0.7	0
0.8	254
0.9	0
1.0	405

n = 879

Mean	0.81	Minimum	0.00
Mode	0.87	Maximum	1.00
Median	1.00	Skewness	-1.17
Std Dev	0.22	Kurtosis	0.56

The mean S0 score exceeds SP (self-evaluation of performance) by 0.07, or about 10%. Informal observation by this experimenter confirms that this is appropriate, i.e., that most searchers got the "true" answers, but not all did so by a well-executed process. These self-evaluation scores appear to reflect this observation.

5. Composite measure of performance by outcome, CO. This measure is the mean of RE and SO, i.e.,

$$CO = (RE + SO)/2$$

Table 11. Frequency Distribution for CO

CO	f
0.0	0
0.1	9
0.2	15
0.3	10
0.4	31
0.5	26
0.6	31
0.7	69
0.8	103
0.9	215
1.0	370

n = 879

Mean	0.85	Minimum	0.10
Mode	1.00	Maximum	1.00
Median	0.92	Skewness	-1.71
Std Dev	0.20	Kurtosis	2.52

Although the mean value of CO (0.85) is considerably higher than that for process performance (CP = 0.49), there is no basis for a direct comparison of these values, even though they do coincide with this experimenter's expectations.

3.1.3 Overall Performance, PERF

This is the mean of CP and CO, used as an overall measure of merit.

Table 12. Frequency Distribution for PERF

<u>PERF</u>	<u>f</u>
0.0	0
0.1	4
0.2	6
0.3	23
0.4	52
0.5	78
0.6	196
0.7	371
0.8	145
0.9	4
1.0	0

n = 879

Mean	0.64	Minimum	0.10
Mode	0.70	Maximum	0.90
Median	0.67	Skewness	-1.20
Std Dev	0.13	Kurtosis	1.70

3.1.4 Attitude Measures

1. Attitude toward commands, AC.

Table 13. Frequency Distribution for AC

<u>AC</u>	<u>f</u>
0.0	0
0.1	1
0.2	11
0.3	6
0.4	19
0.5	32
0.6	57
0.7	64
0.8	22
0.9	0
1.0	6

n = 218

Table 13--Continued

Mean	0.60	Minimum	0.10
Mode	0.70	Maximum	1.00
Median	0.62	Skewness	-0.50
Std Dev	0.17	Kurtosis	0.64

2. Attitude toward the database, AD.

Table 14. Frequency Distribution for AD

AD	f
0.1	0
0.2	8
0.3	1
0.4	11
0.5	26
0.6	11
0.7	66
0.8	50
0.9	13
1.0	21

n = 207

Mean	0.70	Minimum	0.20
Mode	0.70	Maximum	1.00
Median	0.72	Skewness	-0.64
Std Dev	0.19	Kurtosis	0.43

3. Attitude toward the system as a whole, AS.

Table 15. Frequency Distribution for AS

AS	f
0.0	3
0.1	3
0.2	3
0.3	0
0.4	2
0.5	44
0.6	0
0.7	25
0.8	94
0.9	0
1.0	44

n = 218

Table 15--Continued

Mean	0.75	Minimum	0.10
Mode	0.80	Maximum	1.00
Median	0.78	Skewness	-0.82
Std Dev	0.20	Kurtosis	0.75

4. Composite attitude, AT. This variable is the mean of the nonmissing values of the previous attitude measures.

Table 16. Frequency Distribution for AT

AT	f
0.0	0
0.1	0
0.2	0
0.3	3
0.4	7
0.5	19
0.6	56
0.7	55
0.8	47
0.9	18
1.0	0

n = 205

Mean	0.68	Minimum	0.30
Mode	0.60	Maximum	0.90
Median	0.68	Skewness	-0.37
Std Dev	0.13	Kurtosis	-0.02

5. Correlations among attitude measures. These four measures had the following Pearson correlation coefficients and associated significance probabilities:

	AD	AS	AT
AC	0.253 0.000	0.107 0.059	0.626 0.000
AD		0.446 0.000	0.751 0.000
AS			0.739 0.000

Note: The second figure in each cell is significant.

Henceforth, we shall use only one measure of attitude (AT), the composite score, which correlates well with all its components. Clear distinctions between attitudes toward various system components that seemed apparent during the pilot study are not evident here. Probably, we went too far and asked for overly detailed distinctions to be drawn.

3.1.5 Question Analysis

Questions were composed to fit within three categories of relative difficulty, but the difficulty ultimately is determined by actual user opinion and performance. The variables listed below reflect several approaches to measuring difficulty.

1. Intended question difficulty, QD. This is a summary of the assigned difficulty categories given in section 2.4.

Session	No. Questions at Level			Mean	Difference
	1	2	3		
1	2	1	1	1.50	
2	1	2	1	1.75	0.25
3	1	2	1	2.00	0.25
4	1	1	2	2.25	0.25
5	1	1	2	2.25	0.00

2. Quit frequency, QF. This is the frequency of users quitting before finding any answers to a question. It is measured by counting the number of times users did not record on the sheet handed in at the end of a search any records as constituting part of an answer.

<u>Session</u>	<u>No. Quits</u>
1	10
2	0
3	3
4	2
5	4

Inexperience with the language, i.e., a low session number, seems the major cause of quitting, since more than half the total occurred during session 1.

3. Correlations among question difficulty rate and session number.

	<u>Difficulty</u>	<u>Quit Rate</u>
Session No.	0.970	-0.420
	0.003	0.241
Difficulty		-0.519
		0.185

The high correlation of session number and question difficulty was, of course, intended; it is not an experimental finding. The negative correlation of session number and quit rate tends to confirm the earlier suggestion that experience with the system decreases quit rate. The negative correlation of question difficulty with quit rate tends to confirm that it is not the inherent difficulty of the question that causes quitting, but probably the lack of experience.

3.2 Interrelationships

In this section, we review various interrelationships among variables and attempt to identify major patterns.

3.2.1 Question Analysis

We consider this area first here because of the possibility that measures of performance and attitude should be corrected for question difficulty.

1. Performance as a function of QD.

- a. Process performance and its constituents. The chi-square data in this, and following, sections are based on the complete 2 x 2 array of values of the performance variable and the independent variable, such as QD (question difficulty). We have not included the entire table here. In order to conserve space, we have condensed the table. Only the means of the dependent variables are shown for each value of the independent variable.

(1) Maximum commands (CD) as a function of QD

	QD		
	1	2	3
CD	0.89	0.82	0.79

chi-sq = 245.72, 16 d.f., signif. = 0.000

(2) Advanced commands (PA) as a function of QD

	QD		
	1	2	3
PA	0.25	0.34	0.40

chi-sq = 93.71, 20 d.f., signif. = 0.000

(3) Self-evaluation of process (SP) as a function of QD

	QD		
	1	2	3
SP	0.82	0.66	0.74

chi-sq = 105.10, 8 d.f., signif. = 0.000

(4) Precision (PR) as a function of QD

	QD		
	1	2	3
PR	0.13	0.08	0.15

chi-sq = 195.68, 12 d.f., signif. = 0.000

(5) Composite process measure (CP) as a function of QD

	QD		
	1	2	3
CP	0.52	0.46	0.50

chi-sq = 58.09, 14 d.f., signif. = 0.000

- (6) Discussion: All five of these relationships are fairly strong, as measured by chi-square. CD shows a decrease in number of commands used, by question difficulty, whereas the opposite might have been expected. Perhaps what happened was that users of all languages chose to go with L1 when dealing with low difficulty questions, and this language, although simple, is "wordy." L1 requires more commands (choices) to retrieve a given record than L2 or L3 if used effectively.

PA confirms the expected behavior, i.e., a steady increase with QD, indicating that users did go to the more complex and powerful languages as the question seemed to demand.

The SP figures are erratic, perhaps showing only that people did indeed feel they had done well with the easier questions.

PR also confirms erratic behavior, with the medium difficulty questions causing the most trouble. No explanation of the SP and PR behavior is suggested.

CP shows, as expected by the mathematics, the same erratic pattern as SP and PR.

b. Outcome and its constituents.

(1) Recall as a function of QD

RE as a function of QD

	QD		
	1	2	3
RE	0.96	0.87	0.84

chi-sq = 227.44, 12 d.f., signif. = 0.000

(2) Self-evaluation of outcome as a function of QD

SO as a function of QD

	QD		
	1	2	3
SO	0.90	0.77	0.76

chi-sq = 95.50, 10 d.f., signif. = 0.000

(3) Outcome as a function of QD

CO as a function of QD

	QD		
	1	2	3
CO	0.93	0.82	0.78

chi-sq = 190.04, 18 d.f., signif. = 0.000

- (4) Discussion: Here, again, we have a series of strong relationships, all tending to demonstrate a high value for easier questions and roughly equal values for the two more difficult classes of question; hence, CO shows the expected behavior, decreasing with increasing QD.

2. Attitude as a function of QD.

a. AT as a function of interface experience, IX

Table 17. Frequency Distribution of AT as a Function of IX

AT	Session				
	1	2	3	4	5
0.0	0	0	0	0	0
0.1	0	0	0	0	0
0.2	0	0	0	0	0
0.3	0	0	0	1	2
0.4	0	1	3	2	1
0.5	4	4	3	5	3
0.6	11	8	14	9	14
0.7	11	14	10	12	8
0.8	14	10	6	8	9
0.9	2	4	4	5	3

chi-sq = 19.52, 24 d.f., signif. = 0.724

	Session				
	1	2	3	4	5
Mean AT	0.71	0.70	0.66	0.70	0.66
by session.					

- b. AD (attitude corrected for question difficulty) as a function of IX. The correction factor used here was 1.5/DIFF, where 1.5 was the mean question difficulty in session 1 and DIFF the mean in the session for which attitude was recorded.

Table 18. Frequency Distribution of AD as a Function of IX

AD	Session				
	1	2	3	4	5
0.1	0	0	0	0	0
0.2	0	0	0	3	2
0.3	0	0	3	5	4
0.4	0	3	12	16	16
0.5	4	10	15	13	15
0.6	11	17	9	5	3
0.7	11	9	1	0	0
0.8	14	2	0	0	0
0.9	2	0	0	0	0

chi-sq = 141.62, 28 d.f., signif. = 4.000

	Session				
	1	2	3	4	5
Mean AD by session.	0.71	0.59	0.48	0.43	0.40

Discussion: The uncorrected attitude scores, AT, do not show a particularly strong pattern, but the corrected ones do. This supports the suggestion that question difficulty affects attitude toward the system.

Conclusion 1

The variable QD, assigned question difficulty, is an ordinal, not a ratio variable, i.e., it is not a numeric measure. Hence, we must avoid the temptation to use it to correct all performance and attitude measures. QR, the quit rate (see section 3.1.5), does not alone reflect difficulty. The high value for session 1, followed by lower and gradually increasing values, would seem to indicate learning during the initial session, followed by a quit rate that does seem in keeping with the intended increase in difficulty.

We would prefer not to try to fashion a correction factor out of these QD and QR data, but rather to bear in mind that, when interpreting performance and attitude data, a flat curve, over time, represents no change in the face of increasing difficulty and, hence, is indicative of a change of undetermined magnitude. Nonetheless, the analysis of attitude, as a function of session number adjusted for question difficulty, shows a distinct pattern (a steadily decreasing difference between successive attitude scores) as session number increases. This apparently indicates that the attitude levels were approaching a stable level.

3.2.2 Attitude Analysis

1. Attitude as a function of prior experience. Based on prior experience alone, attitude did not show great variation.

Table 19. AT as a Function of Prior Experience (PX)

	PX		
	1	2	3
Mean	0.65	0.69	0.69
Median	0.67	0.69	0.70

2. Attitude as a function of interface complexity. Language 2 users had the strongest positive attitude overall; Language 1 users showed the lowest attitude.

Table 20. AT as a Function of Interface Complexity (IC)

	IC		
	L1	L2	L3
Mean	0.62	0.75	0.68
Median	0.61	0.77	0.68

3. Attitude as a function of prior experience and interface complexity. Here, we can see some distinct patterns. The preference for L2 by E2 (moderately experienced) and E3 (experienced) users stands out, as do the more extreme views of the E3 user group. Note that attitude toward L1 rises slightly with experience and then drops sharply, while attitudes toward L2 and L3 rise steadily with experience, with L2 rising faster than L3 over the time span studied.

Table 21. Attitude by PX and IC

	PX		
	E1	E2	E3
L1	0.69	0.72	0.63
IC L2	0.71	0.77	0.88
L3	0.72	0.72	0.74

4. Attitude as a function of interface system experience (IX or session no.). This relationship, which we had hoped would show a clear pattern, shows instead very little pattern. Various chi-square tests on individual curves and t-tests between curves show no significant relationships or differences.

Table 22. AT as a Function of PX, IC, and IX (or Session Number)

	PX=1					PX=2					PX=3				
	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
L1	76	71	70	65	63	83	72	72	70	60	71	62	58	63	60
IC L2	69	79	70	68	70	79	80	71	74	81	83	89	90	90	87
L3	70	77	71	72	70	73	70	71	73	76	69	79	73	74	74

All figures are x 100.

If we consider only the movement of attitude between the first and last (fifth) sessions, we get rather striking differences. The individual differences in attitude at the first and last sessions fail formal tests of significance. However, if we further aggregate the data, showing only combined attitude for a language user group, as here

	Session	
	1	5
L1	0.76	0.62
L2	0.78	0.81
L3	0.71	0.73

then the drop in attitude toward L1 is significant, but the other changes are not.

5. Attitude as a function of question difficulty. We analyzed attitude, corrected and uncorrected by question difficulty, in section 3.2.1.2.

Conclusions 2

As a whole, users preferred L2, the Language of intermediate complexity.

Conclusion 3

The intensity of opinion toward a language (highest and lowest attitude scores) increases with the amount of prior experience.

Conclusion 4

The attitude toward L1 decreased with experience in its use on the part of all subjects. This trend is statistically significant. The attitudes toward L2 and L3 slightly increased, but these statements are without statistical significance. Formally, we must conclude that there was no difference, over time, between these two (L2,L3) user groups.

Conclusion 5

Prior experience, combined with language assigned, exerted significant influence on attitude. The more complex languages were liked more by the more experienced users, and the most experienced group showed the strongest dislike for the least complex, least powerful language. L1 gives us the only picture of an attitude curve changing direction: it increases from E1 (inexperienced) to E2 users, then decreases from E2 to E3 users. Possibly, this means that, with enough experience, the attitudes toward L2 and L3 would turn down similarly. This was one of our initial assumptions.

3.2.3 Performance Analysis--Process

1. Process performance and its constituents as functions of prior experience.

CD (maximum commands) as a function of prior experience (PX)

	PX		
	1	2	3
CD	0.83	0.83	0.84

chi-sq = 14.71, 16 d.f., signif. = 0.546

PA (percent of advanced commands) as a function of prior experience (PX)

	PX		
	1	2	3
PA	0.24	0.31	0.42

chi-sq = 77.25, 20 d.f., signif. = 0.000

SP (self-evaluation of process) as a function of PX

	PX		
	1	2	3
SP	0.73	0.74	0.75

chi-sq = 4.52, 8 d.f., signif. = 0.808

PR (precision) as a function of PX

	PX		
	1	2	3
PR	0.10	0.12	0.15

chi-sq = 25.16, 12 d.f., signif. = 0.014

CP (composite process measure) as a function of PX

	PX		
	1	2	3
CP	0.47	0.48	0.52

chi-sq = 45.30, 14 d.f., signif. = 0.000

PA and PR show strong relationships with PX, both increasing with increasing PX. CD and SP are virtually constant with respect to PX.

2. Process performance and its constituents as a function of interface complexity(IC).

CD as a function of IC

	IC		
	L1	L2	L3
CD	0.81	0.85	0.84

chi-sq = 47.79, 16 d.f., signif. = 0.000

PA as a function of IC

	IC		
	L1	L2	L3
PA	0.00*	0.33	0.32

chi-sq = 13.93, 10 d.f., signif. = 0.176

* L1 users did not have the option of using L2 or L3

SP as a function of IC

	IC		
	L1	L2	L3
SP	0.71	0.75	0.77

chi-sq = 34.66, 8 d.f., signif. = 0.000

PR as a function of IC

	IC		
	L1	L2	L3
SP	0.09	0.14	0.13

chi-sq = 29.46, 12 d.f., signif. = 0.003

CP as a function of IC*

	IC		
	L1	L2	L3
CP	0.50	0.49	0.48

* There are several different ways of computing CP, given that PA is undefined for L = 1, but all share virtually equal values for CP as a function of IC.

chi-sq = 90.55, 14 d.f., signif. = 0.000

3. Process performance and its constituents as functions of interface system experience.

	Session				
	1	2	3	4	5
CD	0.84	0.86	0.83	0.82	0.82

chi-sq = 80.07, 32 d.f., signif. = 0.000

	Session				
	1	2	3	4	5
PA	0.23	0.32	0.33	0.40	0.34

chi-sq = 118.46, 40 d.f., signif. = 0.000

	Session				
	1	2	3	4	5
SP	0.71	0.77	0.74	0.78	0.72

chi-sq = 36.92, 16 d.f., signif. = 0.000

	Session				
	1	2	3	4	5
PR	0.08	0.16	0.08	0.16	0.11
chi-sq = 177.83, 24 d.f., signif. = 0.000					

	Session				
	1	2	3	4	5
CP	0.45	0.52	0.48	0.52	0.48
chi-sq = 90.44, 28 d.f., signif. = 0.000					

The means show a fluctuation for four out of five of these variables in which we have relatively low values (in sessions 1, 3, and 5, higher in 2 and 4). This particular pattern will be seen again in other measures. These data are based on the means. The medians have a more stable behavior. For CP, the medians by session number are:

.4 .5 .5 .5 .5

Table 23. Frequency Distribution of CP by Session

	Session				
CP	1	2	3	4	5
0.0	0	0	0	0	0
0.1	2	1	0	0	0
0.2	7	0	0	0	1
0.3	21	5	12	7	14
0.4	58	39	58	40	54
0.5	51	70	60	63	64
0.6	28	45	42	38	36
0.7	9	13	3	26	5
0.8	0	3	1	1	2
0.9	0	0	0	0	0

chi-sq = 90.44, 28 d.f., signif. = 0.000

4. Process performance (CP) as a joint function of prior experience (PX), interface complexity (IC), and interface experience.

Table 24. CP by PX and IC

		PX	
	E1	E2	E3
L1	0.49	0.50	0.51
IC L2	0.45	0.47	0.56
L3	0.46	0.49	0.51

chi-sq = 167.47, 56 d.f., signif. = 0.000

This is similar to the plot of attitudes versus PX and IC. The high score is for E3 users of L2. E1 users perform at a lower level than E2, who are lower than E3. The differences, by language, are slight.

Table 25. CP as a Function of PX, IC, and IX (or Session Number)

	PX=1					PX=2					PX=3				
	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
L1	47	50	50	53	47	46	51	51	49	52	46	56	53	55	47
IC L2	43	49	46	50	39	42	48	46	52	48	54	59	50	58	58
L3	41	48	46	49	45	44	52	45	52	50	48	53	50	55	51

All figures are x 100

Session Chi-square df Significance

1	68.82	56	0.026
2	69.44	48	0.023
3	59.75	40	0.023
4	39.25	40	0.504
5	89.30	48	0.000

5. Pearson correlation coefficients among process variables.

	PA	SP	PR	CP
CD	-0.022 0.253	-0.385 0.001	-0.167 0.001	-0.447 0.001
PA		0.054 0.066	0.323 0.001	0.364 0.001
SP			0.337 0.001	0.707 0.001

There is a fairly high correlation between number of commands (CD) and self-perception of process and with the composite process variable. There is a higher correlation between self-perception and the composite score.

Conclusion 6

The E1 and E2 groups performed about the same on the composite measure; the E3 group did significantly better.

Conclusion 7

There was no significant performance difference between the various language groups.

Conclusion 8

Performance, generally, did not change much with time if we do not correct for question difficulty. We may assume, then, that increases in performance, which normally can be expected to come with experience, were just about balanced by increased question difficulty.

3.2.4 Performance Analysis--Outcome

1. Outcome and its constituents as a function of prior experience, (PX).

a. Recall (RE) as a function of PX

	PX		
	1	2	3
RE	0.84	0.90	0.93

chi-sq = 24.94, 12 d.f., signif. = 0.015

b. Self-evaluation of outcome (SO) as a function of PX

	PX		
	1	2	3
SO	0.81	0.81	0.82

chi-sq = 10.51, 10 d.f., signif. = 0.397

c. Outcome (CO) as a function of PX

	PX		
	1	2	3
CO	0.82	0.85	0.87

chi-sq = 28.05, 18 d.f., signif. = 0.061

RE shows a progressive increase with PX, although weakly established. SO shows no meaningful relationship--the means are virtually constant, but chi-square is low. CO, as the mean of these two variables, shows a behavior that is a little of each.

2. Outcome and its constituents as a function of interface complexity, (IC).

a. Recall (RE) as a function of IC

	IC		
	L1	L2	L3
RE	0.92	0.85	0.90

chi-sq = 24.43, 12 d.f., signif. = 0.013

b. Self-evaluation of outcome (SO) as a function if IC

	IC		
	L1	L2	L3
SP	0.79	0.81	0.84

chi-sq = 21.67, 10 d.f., signif. = 0.017

c. Outcome (CO) as a function of IC

	IC		
	L1	L2	L3
CO	0.85	0.83	0.86

chi-sq = 19.88, 18 d.f., signif. = 0.340

Here, we see a contrast. RE decreases with IC, while SO increases, both relations significant within 5%. The mean, CO, fluctuates, and the strength of its relation with IC is not significant.

3. Outcome and its constituents as a function of interface system experience, (IX or session no.)

a. Recall as a function of IX

	Session				
	1	2	3	4	5
RE	0.92	0.92	0.85	0.95	0.82

chi-sq = 176.47, 24 d.f., signif. = 0.000

b. Self-evaluation of outcome as a function of IX

	Session				
	1	2	3	4	5
SO	0.84	0.85	0.81	0.81	0.76

chi-sq = 65.00, 20 d.f., signif. = 0.000

c. Outcome as a function of IX

	Session				
	1	2	3	4	5
CO	0.88	0.88	0.82	0.88	0.78

chi-sq = 94.57, 36 d.f., signif. = 0.000

	Session				
	1	2	3	4	5
CO	0.0	0	0	0	0
	0.1	4	0	2	3
	0.2	1	1	8	1
	0.3	2	1	3	0
	0.4	4	4	7	1
	0.5	3	5	8	3
	0.6	7	4	7	4
	0.7	7	12	11	17
	0.8	15	19	12	35
	0.9	39	53	45	51
	1.0	0	0	0	0

chi-sq = 94.57, 36 d.f., signif. = 0.000

These relationships are perplexing. They have significant chi-square values and a tendency to show a drop in value at sessions 3 and 5. These dips were also apparent in the process performance figures in section 3.2.3.3.

4. Outcome (CO) as a function of prior experience (PX), interface complexity (IC), and system use experience.

Table 26. CO by PX and IC

		PX	
	E1	E2	E3
L1	0.89	0.83	0.82
IC L2	0.78	0.83	0.89
L3	0.80	0.89	0.91

chi-sq = 127.46, 72 d.f., signif. = 0.000

Here is a fairly strong delineated pattern. The best performances are seen to occur with the highest combinations of language level and prior experience--E2-L3, E3-L2, and E3-L3--and with the lowest combination--E1-L1.

Table 27. CO as a Function of PX, IC, and IX (or Session Number)

	PX=1					PX=2					PX=3				
	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
L1	85	93	90	90	89	89	93	83	83	68	86	89	77	87	73
IC L2	77	83	69	85	75	87	88	74	89	77	92	91	93	89	80
L3	83	77	77	87	76	95	91	86	89	80	97	92	93	90	84

Session	Chi-square	df	Significance
1	88.13	72	0.093
2	64.46	64	0.466
3	83.98	72	0.158
4	56.20	56	0.467
5	62.05	72	0.794

This is the PX-IC data extended to include session number as a variable. Broken down to this level of detail, the relationships are not significant.

5. Pearson correlation coefficients among outcome variables.

	<u>RE</u>	<u>SO</u>
RE	0.302 0.001	0.854 0.001
SO		0.750 0.001

Conclusion 9

There is a small increase in performance measures as a function of prior experience. This experiment did not continue long enough to permit detection of the point where the increase with experience ends, if at all.

Conclusion 10

Measures of performance as a function of interface complexity or language are alone inconclusive. There is, however, a slight edge for L1 and L3.

Conclusion 11

E1 users did much better with L1 than the others did. E2 and E3 users did much better with L2 and L3 than the others did.

Conclusion 12

E1 users did worse with increasing interface complexity. E2 and E3 users did better with increasing interface complexity, although all these trends are slight. This may indicate that the E1 group was able to handle L1 and to be challenged by it, while the others had more experience than L1 called for and, thus, their negative attitudes were reflected in poorer performance. That E2 and E3 subjects did better with increasing language complexity may indicate that these languages were better suited to their own cognitive complexity.

Conclusion 13

There is a very slight, but not statistically significant, tendency for outcome to decrease as a function of time or to fail to change with time. Again, this is with outcome data uncorrected for question difficulty. It may be an indication of an actual increase (of unknown magnitude) against increasingly difficult questions. The mysterious drop in values during session 5 is still apparent, and its implication unknown.

3.2.5 Overall Composite Performance

The variable PERF is the mean of the two other performance composites--process and outcome. PERF distributions are:

1. PERF as a function of PX.

	PX		
	1	2	3
PERF	0.64	0.67	0.70

chi-sq = 34.44, 16 d.f., signif. = 0.005

We see here a neat linear growth of PERF as a function of PX.

2. PERF as a function of IC.

	IC		
	L1	L2	L3
PERF	0.68	0.66	0.67

chi-sq = 24.09, 16 d.f., signif. = 0.088

This nearly constant level of PERF with IC is not significantly strong as a relationship.

3. PERF as a function of PX and IC.

	PX		
	1	2	3
L1	0.69	0.67	0.67
IC L2	0.62	0.65	0.72
L3	0.63	0.69	0.71

chi-sq = 114.54, 64 d.f., signif. = 0.000

Here, the high performance of the high E-L combinations at E1-L1 is maintained.

4. PERF by session no.

	Session				
	1	2	3	4	5
PERF	0.67	0.70	0.65	0.70	0.63

chi-sq = 74.82, 32 d.f., signif. = 0.000

The same dips at sessions 3 and 5 continue to perplex.

5. PERF as a function of PX, IC, and IX.

	PX=1					PX=2					PX=3				
	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
L1	77	82	83	83	79	80	83	80	78	72	77	83	76	82	69
IC L2	67	72	64	74	64	70	74	67	77	69	81	82	80	80	76
L3	69	70	68	74	68	77	78	73	77	72	79	79	79	79	75

All figures are x 100

Session	Chi-square	df	Significance
1	65.06	56	.190
2	59.8	48	.124
3	56.55	48	.186
4	35.77	40	.661
5	65.04	64	.446

Conclusion 14

Overall performance does seem to increase with increasing prior experience, but the implications of language used or search number at this level of detailed analysis are inconclusive.

3.2.6 Correlation of Attitude with Performance

The Pearson r correlation coefficient between PERF and AT is .162 with a significance of .143, indicating a low degree of correlation. This was based on a correlation of the means of the two measures broken down by PX, IC, and IX. Since neither of these variables shows a wide range of variation and both suffer from lack of precise definition, it was not clear what, if any, conclusion should be drawn from this value of r .

3.2.7 Performance Corrected for Question Difficulty

Our final analysis returns to the use of relative question difficulty as a correction factor. Figures presented below show the three composite scores--CP, CO, and PERF--corrected for question difficulty and displayed as functions of PX, IC, and IX. The correction used was to multiply each individual score on a search by the difficulty (1,2, or 3) of the question involved. This, of course, gives a resultant range of 0 to 3, rather than our traditional 0 to 1. The significance of the chi-square measures tends to be less (higher probability of random occurrence of the distribution) than for some of our earlier analyses.

1. CP, CO, and PERF as functions of prior experience.

	PX		
	E1	E2	E3
CP	0.90	0.94	1.02
CO	1.55	1.60	1.65
PERF	1.21	1.26	1.32

CP: chi-sq = 57.01, 30 d.f., signif. = .002 *
CO: chi-sq = 35.80, 38 d.f., signif. = .572 *
PERF: chi-sq = 61.77, 46 d.f., signif. = .060 *

* Again chi-square is calculated
from a complete table of independent
variable values, not this array of means.

Each line shows a steady, monotonic increasing performance gain as a
function of prior experience.

2. CP, CO, and PERF as functions of IC.

	IC		
	L1	L2	L3
CP	0.97	0.94	0.94
CO	1.61	1.56	1.64
PERF	1.27	1.23	1.27

CP: chi-sq = 292.24, 60 d.f., signif. = 0.000
CO: chi-sq = 270.94, 76 d.f., signif. = 0.000
PERF: chi-sq = 221.52, 92 d.f., signif. = 0.000

The differences among languages are not pronounced, but, in each category,
L2 users perform at a lower level than L1 and never better than L3. L1
and L3 are about the same.

3. CP, CO, and PERF as functions of IX.

	Session				
	1	2	3	4	5
CP	0.66	0.90	0.95	1.16	1.09
CO	1.30	1.49	1.58	1.94	1.71
PERF	0.97	1.17	1.25	1.53	1.38

chi-sq = 292.24, 60 d.f., signif. = 0.000

chi-sq = 270.94, 76 d.f., signif. = 0.000

chi-sq = 221.52, 92 d.f., signif. = 0.000

We continue to see the slump in session 3, dampened by the correction factor now to show an increase in session 3 over session 2, but by a smaller amount than the S2-S1 or S4-S3 differences. Session 5 continues to show an absolute decrease in performance.

4. CP, CO, and PERF as functions of PX and IC.

CP:

	PX		
	1	2	3
L1	0.94	0.96	0.99
IC L2	0.87	0.92	1.07
L3	0.88	0.94	1.00

chi-sq = 211.74, 120 d.f., signif. = 0.000

CO:

	PX		
	1	2	3
L1	1.70	1.55	1.56
IC L2	1.44	1.57	1.69
L3	1.51	1.68	1.73

chi-sq = 167.22, 152 d.f., signif. = .189

PERF:

		PX	
	1	2	3
L1	1.31	1.24	1.26
IC L2	1.14	1.23	1.36
L3	1.18	1.29	1.31

chi-sq = 223.71, 184 d.f., signif. = .023

Conclusion 15

Performance, on all three composite measures, improves with prior experience, i.e., for a given task and level of interface complexity, those with more prior computing experience will perform better.

Conclusion 16

We cannot attribute performance improvement to interface complexity or language alone.

Conclusion 17

Performance generally improves with experience in use of the system, but a drop in all composite scores in the fifth session persists throughout the data. We cannot tell whether this is some negative learning behavior, the questions were more difficult than intended, or the subjects were tiring of the experiment. Other explanations are no doubt possible.

Conclusion 18

It is reasonable to conclude that the matching of users with languages does affect performance. Inexperienced users consistently did well with the least complex of the languages. Experienced users consistently did well with the more complex languages. Inexperienced users did poorly with complex languages, and, perhaps what is most significant, experienced users did poorly with the least complex language.

4. OVERALL CONCLUSIONS

4.1 Restatement of Assumptions

We expected that:

1. Performance of users is a function of both prior experience and complexity of the language used--not in a monotonic relationship such that more of either or both independent variables improve performance, but that performance level is determined by the best fit of language level to user skill.
2. Attitude toward a new language (essentially, to most users, "language" = "system") will rise as learning takes place, then decline as users achieve a high level of skill and become interested in problems that the system does not solve well. Attitude was predicted as a concave downward curve, with time, but the starting point for any user, or the span of time needed to rise to the maximum and then recede, may vary considerably, depending on user prior experience and language complexity (power).
3. Experience using the language being learned will, with time, be added to prior experience, so that more experience with the test system will result in increased levels of performance. This, of course, is the traditional view that practice improves performance.

The implications of these assumptions, if true, are that:

1. No one language fits all users. On the contrary, languages should be designed to fit the needs and skills of identified classes of users.
2. Users change with time, in respect to both attitude and performance, but the rate of change is not uniform. It depends on the prior experience and the interface complexity. Assigning the wrong language to a user may inhibit performance. This is true if either the language is too complex for the user's skill level or insufficiently powerful for his abilities.

4.2 Conclusions

We do not claim to have proven any of these assumptions, but the data suggest that:

1. Performance generally, but not always, improves with experience, a conclusion that is unsurprising (section 3.2.5).
2. Attitude toward the less complex languages decreased over time (3.2.2), while it increased for the higher order (more complex, powerful) languages. This would seem to imply that users' experience with the test system was increasing their overall skill levels and their perception of language capability.
3. Attitude does not show the expected parabolic curve with time (3.2.3), nor did it correlate highly with performance (3.2.6). We suspect our own measuring procedures were at fault. In retrospect, we would have preferred to begin attitude measurement immediately at the end of the training sessions, instead of at the end of the first search session.
4. The intensity of attitude--positive or negative--is strongly influenced by prior experience (3.2.2), indicating that the user population will not have a uniform attitude toward a language, and that less skilled users, not unexpectedly, will have less well-formed opinions about the language being used.
5. Performance, considered solely as a function of language used, shows little variation. One must look beyond language complexity to predict performance (3.2.3.2, 3.2.4.2, 3.2.5.2).
6. Since the best performances were achieved by the least experienced group using the least complex language and the most experienced groups using the most complex languages (3.2.3.4, 3.2.4.4, 3.2.5.3), we again conclude that it does matter what language or language subset a user is assigned to work with.
7. Prior experience seemed to be the most consistent user-related factor in predicting performance (3.2.2.1, 3.2.5.1). Question difficulty, or environmental complexity, was also a factor of importance (3.2.1).

It is regrettable that the report of almost any experiment in this field concludes that more research is needed--that none of us claim to have resolved the issue under study. This is due in part to the expense and complexity of such studies and the innate difficulty of identifying and measuring the appropriate variables. In the case of the study at hand, we feel that a repetition with a larger sample over a considerably longer period of time would be valuable. Perhaps the extended time would enable slight trends to be seen as strong movements.

In spite of known difficulties in the collection and interpretation of these data, we feel that something of value has been accomplished in the struggle to understand the interrelation of users and computer systems.

REFERENCES

- Anderson, John R. 1980. Cognitive psychology and its implications, 227. San Francisco: W.H. Freeman.
- Carlisle, James H. 1974. Man computer interactive problem solving: Relationships between user characteristics and interface complexity. New Haven, CT: Yale University School of Organization and Management. Technical report no. 77. Available from NTIS, Springfield, VA: AD 786466.
- Dolby, James L. 1971. Progress in documentation: Programming languages in mechanized documentation. Journal of Documentation 27(2): 136-55.
- Fenichel, Carol Hansen. 1979. Online information retrieval: Identification of measures that discriminate among users with different levels and types of experience. Ph.D. diss., School of Library and Information Science, Drexel University.
- Hewett, Thomas T., and Charles T. Meadow. 1981. A study of the measurement of user performance, 49-76. In Public access to library automation. Ed. J.L. Divilbiss. Urbana, IL: Graduate School of Library and Information Science, University of Illinois.
- King, Donald, and E.C. Bryant. 1971. The evaluation of information services and products, 20-33. Washington: Information Resources Press.
- Lodge, H. 1980. Investigation into the online searching techniques used in a group of British university libraries, 64-79. Ph.D. diss., Postgraduate School of Librarianship and Information Science, University of Sheffield, Sheffield, England.
- Marcus, Richard S., and J. Francis Reintjes. 1981a. A translating computer interface for end user operation of heterogeneous retrieval systems. I. Design. Journal of the American Society for Information Science 32(4): 287-303.
- 1981b. A translating computer interface for end user operation of heterogeneous retrieval systems. II. Evaluations. Journal of the American Society for Information Science 32(4): 304-317.
- Meadow, Charles T. 1981. Matching users and user languages in information retrieval. Online Review 5(4): 313-22.
- Meadow, Charles T., Thomas T. Hewett, and Elizabeth S. Aversa. 1982a. A computer intermediary for interactive database searching. I. Design. Journal of the American Society for Information Science 33(5): 325-32.
- 1982b. A computer intermediary for interactive database searching. II. Evaluation. Journal of the American Society for Information Science 33(6): 357-64.
- Mooers, Calvin N. 1960. Mooers Law or, why some retrieval systems are used and others are not. American Documentation 11(3): ii.

Nie, Norman H. et al. 1975. Statistical package for the social sciences. 2d ed. New York: McGraw-Hill.

Penniman, William David. 1975. Rhythms of dialogue in human-computer communication. Ph.D. diss., The Ohio State University, Columbus.

Wanger, Judith, and Dennis McDonald. 1980. Evaluation of the online process. Santa Monica, CA: Cuadra Associates. Report no. NLM 78-7. Available from NTIS, Springfield, VA: PB 81-132565.

Walker, Donald E. 1981. The organization and use of information: Contributions of information science, computational linguistics and artificial intelligence. Journal of the American Society for Information Science 32(5): 347-63.

Whorf, Benjamin J. 1956. Language, thought, and reality: Selected writings, vi. Ed. J.B. Carroll. Cambridge, MA: MIT Press.

APPENDIX A. Reference File

This is the text of an online reference "library" that was available to users through the HELP command. It was not intended to be read in its entirety, but to be consulted, when necessary, for help with a specific problem. This text is organized in a hierarchical manner. Any record can have a subordinate, in the sense that a subordinate is a logical next record for the one being read. The parent of a record is the predecessor. A record can have up to five subordinates (possible successors) but only one predecessor. The level number indicates the depth in the hierarchy. The first record is at level 1; one of its successors is at level 2, etc. At the conclusion of each record is a list of possible successors. The number in parentheses is the record number, the first element in each record. There is no fixed order of reading this text. Thus, from record no. 1, if you want to learn how to use HELP, go to record no. 2. If you want to learn about Searching, go to record no. 4. The intent is that a reader would start at the beginning and then follow any subject of interest. It is not intended to be read exhaustively, but to be used for occasional reference.

*** 001, PARENT = 000, LEVEL = 001, NO. SUBORDINATE RECDs = 003

This is the HELP facility. It contains information about how to use the system. If you have not used HELP before select choice no. 1. Otherwise follow the subject headings of interest. Details on the use of specific commands can be found under SEARCHING.

1. (002) How to use HELP
2. (003) Database content and structure
3. (004) Searching
4. (...) QUIT

*** 002, PARENT = 001, LEVEL = 002, NO. SUBORDINATE RECDs = 001

There is a file of HELP records that tell how to use the search system. If the text of any record is too long to fit into a single page, then the record ends with the notation ..CONTINUE. This means the next record is a continuation of the one you are reading and you should select that option.

..CONTINUE

1. (005) More about HELP
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 003, PARENT = 001, LEVEL = 002, NO. SUBORDINATE RECDs = 001

The database contains a set of condensed news articles organized by subject. Subject headings are found in INDEX RECORDS which are also part of the database. Any subject heading may be subdivided into other subject headings--making a hierarchical structure averaging 6 to 8 levels in depth.

..CONTINUE

1. (006) Types of records
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 004, PARENT = 001, LEVEL = 002, NO. SUBORDINATE RECDs = 001

There are two basically different ways to search the database. They may be used separately or in combination.

The first method is called CHOICE-OF-PATH. You begin at the root and follow the appropriate subject subdivisions downward until you find the data records of interest.

..CONTINUE

1. (014) More about searching
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 005, PARENT = 002, LEVEL = 003, NO. SUBORDINATE RECDs = 001

You can search the HELP file using the choice-of-path method in which you select any option offered at the end of a page. You CANNOT use commands in HELP.

If you select RETURN TO ROOT, you go to the root of the HELP file.

If you select RETURN TO SEARCH, you will be asked to confirm that you want to return and will then see the last record retrieved before you asked for HELP.

..CONTINUE

1. (066) More HELP
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 006, PARENT = 003, LEVEL = 003, NO. SUBORDINATE RECDs = 003

The database consists of BOTH the data records and subject-defining index records. It is somewhat as if the books of a library were condensed and filed in the card catalog so that books and subject headings were intermixed. There is also a dictionary listing the words that occur in either data or index records.

1. (007) Index records
2. (008) Data records
3. (060) Dictionary
4. (001) RETURN TO ROOT
5. (...) QUIT

*** 007, PARENT = 006, LEVEL = 004, NO. SUBORDINATE RECDs = 002

An index record serves only to define subject matter. Index records POINT to other records which may be index or data records. The very first index record is called the ROOT and differs slightly from the others.

1. (009) Root index record
2. (010) Other index records
3. (001) RETURN TO ROOT
4. (...) QUIT

*** 008, PARENT = 006, LEVEL = 004, NO. SUBORDINATE RECDs = 001

The data records contain the substance of the database. They are condensed from articles in British newspapers of Oct 1980 - March 1981. Spelling of words is as found in the original item and may vary from item to item. For example there are two common spellings of the Iraqi capital, "Bagdad" and "Baghdad", and both are found in the database.

..CONTINUE

1. (012) More about data records
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 009, PARENT = 007, LEVEL = 005, NO. SUBORDINATE RECDs = 000

The root is the first index record and has no other record pointing to it. It defines the first divisions of the subject matter of the database.

1. (001) RETURN TO ROOT
2. (...) QUIT

Date: 1983 July 15

*** 010, PARENT = 007, LEVEL = 005, NO. SUBORDINATE RECDs = 001

All index records other than the root have a PARENT record--the one that points down to them in the hierarchy.

All index records also have subordinates (OFFSPRING or CHILDREN), including the root. To repeat--the root has no parent and is the only record in the database that does not. All index records have offspring.

..CONTINUE

1. (011) More about other index records
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 011, PARENT = 010, LEVEL = 006, NO. SUBORDINATE RECDs = 000

Here is an example of an index record.

CHOOSE FROM AMONG THE FOLLOWING:

1. LABOUR
2. INDUSTRY
3. GENERAL ECONOMIC INDICATORS
4. RETURN TO ROOT
5. QUIT

The numbered choices refer to paths open through the database or actions to be taken. They are further described under the subject heading SEARCHING.

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 012, PARENT = 008, LEVEL = 005, NO. SUBORDINATE RECDs = 001

Data records have no offspring. They cannot be subdivided. They ALWAYS have a parent which is always an index record.

..CONTINUE

1. (013) More about data records
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 013, PARENT = 012, LEVEL = 006, NO. SUBORDINATE RECDs = 000

Here is an example of a data record.

ON 23 SEPTEMBER 1980 BSC CLOSED THE CONSETT STEEL WORKS

1. (001) RETURN TO ROOT
2. QUIT

The two numbered lines do not contain data but refer to choices of action. There are no choices of path except to return to the root. This is explained further under the subject heading SEARCHING.

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 014, PARENT = 004, LEVEL = 003, NO. SUBORDINATE RECDs = 001

Following a path is done by selecting from among the choices offered after each index record.

If you reach the wrong data record or find yourself on the wrong subject path, you must return to the root to try a different path.

It is useful to keep pencil and paper handy to help keep track of paths tried.

..CONTINUE

1. (015) Searching by command
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 015, PARENT = 014, LEVEL = 004, NO. SUBORDINATE RECDs = 003

The second search method involves telling the computer what records you want to see instead of accepting its choices. To do this, you give the computer COMMANDS. You can, for example, command it to find all records containing the word SEARCH.

Below is more detail on

1. (016) Choice-of-path method of search
2. (017) Searching with basic commands
3. (018) Advanced commands
4. (001) RETURN TO ROOT
5. (...) QUIT

*** 016, PARENT = 015, LEVEL = 005, NO. SUBORDINATE RECDs = 003

Whenever a record is displayed on the VDU--whether index or data--the choices available are displayed with it. The basic choices are listed below. They are not all available all the time. Following the description of the choice, we tell you when it may or may not appear.

1. (019) Down to an offspring
2. (020) Return to root
3. (022) Quit
4. (001) RETURN TO ROOT
5. (...) QUIT

*** 017, PARENT = 015, LEVEL = 005, NO. SUBORDINATE RECDs = 001

In the choice-of-path mode of searching, you select from among the options offered by the computer. In the command mode of searching, you take the initiative and tell the computer how you want to search. Use of commands can be faster, but involves making more decisions. There are three basic kinds of commands.

..CONTINUE

1. (023) More about basic commands
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 018, PARENT = 015, LEVEL = 005, NO. SUBORDINATE RECDs = 001

If you look upon the basic commands as giving an aided increment of searching capability beyond that offered by the choice-of-path method, then the advanced commands may be thought of as yet another increment. Mostly they are not new kinds of commands, but provide a number of variations on the basic ones.

..CONTINUE

1. (044) More about advanced commands
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 019, PARENT = 016, LEVEL = 006, NO. SUBORDINATE RECDs = 001

Whenever an index record is displayed, you may choose to view any of its immediate offspring. These are always numbered from 1 to a maximum of 5. If there are four offspring of the record you are viewing (the CURRENT record) and you select three, you will next see the third offspring of the current record.

..CONTINUE

1. (024) More about down to an offspring
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 020, PARENT = 016, LEVEL = 006, NO. SUBORDINATE RECDs = 000

Anytime you decide you are on the wrong branch of the hierarchical TREE structure or you want to see different information you can go back to the root and start down again, following a different path.

To avoid confusion, this option is not offered when the root is displayed but is offered with every other index or data record.

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 021, PARENT = 026, LEVEL = 008, NO. SUBORDINATE RECDs = 000

With every record displayed, there is the option to quit the search. You can accomplish exactly the same thing with the command QUIT.

In either case, you will be asked if you meant it--to avoid quitting by accident. Answer YES or NO. If you say NO you can then give another command or choice.

If you do quit, you will be asked two questions about your evaluation of the search.

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 022, PARENT = 016, LEVEL = 006, NO. SUBORDINATE RECDs = 001

The QUIT option terminates a search. After selecting the number corresponding to QUIT, you cannot execute any other option.

To prevent unintentional quitting, the computer asks if you meant to quit, and if you say NO you may then make another choice.

The QUIT option is offered after any kind of record: root - index - data.

..CONTINUE

1. (025) More about quitting a search
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 023, PARENT = 017, LEVEL = 006, NO. SUBORDINATE RECDs = 001

PAGE commands enable you to move through the records of the database with more options than are offered in choice-of-path mode.

FIND enables you to retrieve a record directly, based on some word being found within it, without wending your way through the hierarchy.

DICT enables you to search the dictionary for the best words to use with a FIND command.

..CONTINUE

1. (026) More about basic commands
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 024, PARENT = 019, LEVEL = 007, NO. SUBORDINATE RECDs = 000

There are no offspring of a data record, so the only choices after viewing a data record are:

1. (001) RETURN TO ROOT
2. (...) QUIT

Remember--index records always have subordinates or offspring.
Data records never do.

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 025, PARENT = 022, LEVEL = 007, NO. SUBORDINATE RECDs = 000

At the end of the search, you will be asked two questions about it. They call for you to evaluate how well you think the results you got satisfied the search problem you were working on and how well you think you used the search system to get these results.

These are two different questions: one about results and one about method.

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 026, PARENT = 023, LEVEL = 007, NO. SUBORDINATE RECDs = 004

For specific information about how to use commands, select the number corresponding to the command of interest:

1. (027) PAGE
2. (028) FIND

3. (029) DICT
4. (021) QUIT
5. (001) RETURN TO ROOT
6. (...) QUIT

*** 027, PARENT = 026, LEVEL = 008, NO. SUBORDINATE RECDs = 001

A page is a record of the database or a segment of the dictionary. The command PAGE asks for display of a new page. In its simplest form, PAGE retrieves a record adjacent to the current one in the hierarchy. You may ask for:

The parent

The first offspring

The offspring of the parent preceding the current one

The offspring following the current one

..CONTINUE

1. (030) More about PAGE
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 028, PARENT = 026, LEVEL = 008, NO. SUBORDINATE RECDs = 001

The essential idea of the FIND command is that you can retrieve records independently of their location in the hierarchy. Briefly--

FIND STEEL

will locate every record of the database in which the word STEEL is to be found. Those records that have the word, in this case STEEL, will be put into a SET, and you may then search through the set.

..CONTINUE

1. (035) More about FIND
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 029, PARENT = 026, LEVEL = 008, NO. SUBORDINATE RECDs = 002

The command DICT TERM is used to look up a term in the dictionary. The response is a list of five words with your term in the center. The other words are the two preceding and the two following your term in alphabetical order within the dictionary.

All words are accompanied by their frequencies of occurrence.

..CONTINUE

1. (059) More about the dictionary
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 030, PARENT = 027, LEVEL = 009, NO. SUBORDINATE RECDs = 004

In the general case there are four options: you can go UP, DOWN, RIGHT, or LEFT from the current record. Of course not all options are always available. You cannot go up from the root or down from a data record. Select the detailed explanation of choice:

1. (031) PAGE UP
2. (032) PAGE DOWN
3. (033) PAGE RIGHT
4. (034) PAGE LEFT
5. (001) RETURN TO ROOT
6. (...) QUIT

*** 031, PARENT = 030, LEVEL = 010, NO. SUBORDINATE RECDs = 000

PAGE UP or its abbreviation PAGE U tells the computer to retrieve and display the parent of the current record. The parent is always an index record.

PAGE UP or PAGE U has no meaning if the current record is the root.

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 032, PARENT = 030, LEVEL = 010, NO. SUBORDINATE RECDs = 000

PAGE DOWN or its abbreviation PAGE D tells the computer to retrieve and display the FIRST OFFSPRING of the current record. It has no meaning if the current record is a data record because data records have no offspring.

PAGE DOWN or PAGE D gives exactly the same result as selecting choice number 1 after any index record has been displayed.

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 033, PARENT = 030, LEVEL = 010, NO. SUBORDINATE RECDs = 000

PAGE RIGHT or its abbreviation PAGE R gives you the next sibling of the current record, i.e., the next offspring of the same parent. If you have just seen the first offspring, then PAGE R will get you the second. Another PAGE RIGHT will get the third and so on.

PAGE R after the last offspring gets the first again--it operates cyclically so that next after last is first.

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 034, PARENT = 030, LEVEL = 010, NO. SUBORDINATE RECDs = 000

PAGE LEFT or its abbreviation PAGE L gets you the previous sibling of the current record, i.e., the previous offspring of the parent of the current record. If you have just seen the second offspring, PAGE L gets you the first. Another PAGE LEFT will get you the last offspring because PAGE L, like PAGE R, is cyclical. After the first comes the last.

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 035, PARENT = 028, LEVEL = 009, NO. SUBORDINATE RECDs = 001

There are a few rules to remember about the use of FIND:

The term used with the command must be a single word. You can say FIND STEEL or FIND RAILS, but not FIND STEEL RAILS.

If there were no records containing your term, the result is A NULL SET--there are no records in the set.

Only words that are in the dictionary will give you non-null sets.

..CONTINUE

1. (036) More about FIND and sets
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 036, PARENT = 035, LEVEL = 010, NO. SUBORDINATE RECDs = 001

The reason that only dictionary words give non-null sets is that the dictionary contains all the words in the database (with a few exceptions, such as AND, THE, IN, or any "word" that begins with a number).

..CONTINUE

1. (037) More about FIND and sets
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 037, PARENT = 036, LEVEL = 011, NO. SUBORDINATE RECDs = 001

More facts about FIND:

The only way to create a set is with a FIND command.

There is only one set. When you make a new one, the old one is erased.

If your FIND resulted in a null set, the old set will NOT have been erased.

When you create a new set, its first record is automatically displayed.

..CONTINUE

1. (038) More about FIND and sets
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 038, PARENT = 037, LEVEL = 012, NO. SUBORDINATE RECDs = 002

When we say the FIRST record of a set is displayed, we mean first in the hierarchy. An index record at level 2 will be displayed before a data record at level 8. Careful though--it is possible for a data record to be at a lower level than an index record.

You can view the records of a set using the PAGE SET commands.

1. (039) More about sets
2. (040) PAGE SET commands
3. (001) RETURN TO ROOT
4. (...) QUIT

*** 039, PARENT = 038, LEVEL = 013, NO. SUBORDINATE RECDs = 001

To review the SET concept:

1. A set consists of all records having the term specified by a FIND command.
2. Only a FIND can create a set or erase one.
3. There is only one set. Creation of a new one erases the old one.
4. A FIND that retrieves no records is said to create a NULL Set. In this case the previous set is not ERASED.

..CONTINUE

1. (043) More review of sets
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 040, PARENT = 038, LEVEL = 013, NO. SUBORDINATE RECDs = 002

The PAGE SET commands enable you to call for display of successive records of a set.

To use PAGE SET, you MUST have first created a set using a FIND command.

There are two variations--PAGE SET DOWN (or PAGE SET D) and PAGE SET UP (or PAGE SET U)--permitting you to move down or up the list of records in the set. For more detail select:

1. (041) PAGE SET DOWN (or D)
2. (042) PAGE SET UP (or U)
3. (001) RETURN TO ROOT
4. (...) QUIT

*** 041, PARENT = 040, LEVEL = 014, NO. SUBORDINATE RECDs = 000

PAGE SET DOWN or PAGE SET D gets you the next record in the set after the current one. If you have just seen the first record after a FIND command, then PAGE SET D will get the second. Another PAGE SET D will get the third.

Remember that records are in order by level within the hierarchy.

PAGE SET is cyclical. After the last record of a set another PAGE SET D gets the first record again.

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 042, PARENT = 040, LEVEL = 014, NO. SUBORDINATE RECDs = 000

PAGE SET UP or PAGE SET U gets you the record previous to the current one in the set created by a FIND command. If you have just seen the third one you will get the second.

PAGE SET U is cyclical. After the first record of a set PAGE SET U gets the last one.

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 043, PARENT = 039, LEVEL = 014, NO. SUBORDINATE RECDS = 000

5. When a new set is created, its first record is automatically displayed. If a null set is created the last record viewed is shown again.
6. To view other records of the set, use the PAGE SET D or U command.
7. Records will be displayed in order of their occurrence in the hierarchy.
8. Only use of a term that is in the dictionary will result in a set that is not null.
 1. (001) RETURN TO ROOT
 2. (...) QUIT

*** 044, PARENT = 018, LEVEL = 006, NO. SUBORDINATE RECDS = 004

The advanced commands enable you to do the following (select the type command you want to learn about next):

1. (045) Search for (FIND) the root of a word
2. (046) Combine two sets together
3. (047) Combine more than two sets together
4. (048) Review the history of your search
5. (001) RETURN TO ROOT
6. (...) QUIT

*** 045, PARENT = 044, LEVEL = 007, NO. SUBORDINATE RECDS = 001

There may be several variant forms of a word in the dictionary, such as IRAN-IRANIAN- IRANIANS, that all clearly imply the subject IRAN.

To find all records containing any of these forms you can use the command

FIND IRAN :

The ":" symbol means to find all words that begin with the string of letters preceding. All the retrieved records go into a single set.

..CONTINUE

1. (049) More about truncation
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 046, PARENT = 044, LEVEL = 007, NO. SUBORDINATE RECDS = 001

The command FIND A OR B means to create a set having all records that contain either the term A or the term B or both.

The command FIND A AND B means to create a set of records having BOTH A and B in them.

The command FIND A NOT B means to create a set of records that have term A, but DO NOT have term B.

..CONTINUE

1. (051) More about combining two sets
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 047, PARENT = 044, LEVEL = 007, NO. SUBORDINATE RECDS = 002

There are two FIND variations that enable you to use up to five terms at a time in the definition of a set. But you must connect them ALL with AND or with OR. The commands are:

1. (055) FIND AND TERM1 TERM2 TERM3 TERM4 TERM5
2. (056) FIND OR TERM1 TERM2 TERM3 TERM4 TERM5
3. (001) RETURN TO ROOT
4. (...) QUIT

*** 048, PARENT = 044, LEVEL = 007, NO. SUBORDINATE RECDS = 001

The HIST command gives you a display of the first 18 FIND (i.e., set-forming) commands that were carried out during your search so far. Invalid commands that were rejected are not listed.

If you have done more than 18 FIND commands, you will see the question WANT MORE...(YES NO); if you do--just say YES.

..CONTINUE

1. (058) More about the HIST command
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 049, PARENT = 045, LEVEL = 008, NO. SUBORDINATE RECDS = 001

In other words, to find all the variations of a word, truncate it or enter only its root. You then search for all words beginning with that root regardless of what follows. The ":" identifies your command as a truncated search.

The ":" can only be used at the end of a word--never at the beginning or in the middle.

..CONTINUE

1. (050) More about truncation
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 050, PARENT = 049, LEVEL = 009, NO. SUBORDINATE RECDs = 000

*** Some cautions:

- o A command such as:

FIND I :

may retrieve more records (200) than can fit in the space allowed in computer memory. If this happens, the resulting set will be limited to the first 200 records only.

- o The command:

FIND IRA :

will get all variant forms of IRAN and of IRAQ, but will also retrieve IRATE.

- o There must be a space before the ":" as in

FIND IRAN :

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 051, PARENT = 046, LEVEL = 008, NO. SUBORDINATE RECDs = 001

- o Suppose there is a record in the database that has within it the text
SOCRATES IS A MAN.

The command

FIND SOCRATES OR PLATO will retrieve it.

The command

FIND SOCRATES AND MAN will retrieve it.

The command

FIND SOCRATES AND PLATO will not retrieve it.

The command

FIND SOCRATES NOT PLATO will retrieve it.

..CONTINUE

1. (052) More about combining two sets
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 052, PARENT = 051, LEVEL = 009, NO. SUBORDINATE RECDs = 001

Note:

Leave a space on either side of the AND, OR, or NOT when you use this form of FIND. If you enter a command such as

FIND UK ORBRITAIN

the computer will interpret ORBRITAIN as a single word and assume you have omitted the operator (i.e., AND, OR, or NOT).

..CONTINUE

1. (053) A variation on the two-set FIND command
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 053, PARENT = 052, LEVEL = 010, NO. SUBORDINATE RECDs = 001

By using the special symbol /P/ (for PREVIOUS SET), you can combine a new set with the previously existing one. For example:

FIND UK, followed by FIND /P/ AND ECONOMY, gives exactly the same result as FIND UK AND ECONOMY.

You can continue, entering, which FIND /P/ and THATCHER gives a set of records containing all three words: UK, ECONOMY, and THATCHER.

..CONTINUE

1. (054) More about FIND /P/
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 054, PARENT = 053, LEVEL = 011, NO. SUBORDINATE RECDs = 000

The sequence of commands:

FIND BARCLAYS OR LLOYDS

FIND /P/ AND INTEREST

FIND /P/ AND LOANS

should recover any record containing both the words INTEREST and LOANS as well as either bank name, BARCLAYS or LLOYDS.

You may use /P/ only in the first position following a FIND. For example:

FIND /P/ OR UK is good

FIND UK OR /P/ is not good.

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 055, PARENT = 047, LEVEL = 008, NO. SUBORDINATE RECDs = 000

With the FIND AND command, you may use up to five terms. You may use fewer than five, but not less than two.

This command is the equivalent of the sequence:

FIND TERM1 AND TERM2

FIND /P/ AND TERM3

FIND /P/ AND TERM4

FIND /P/ AND TERM5

but the FIND AND command is quicker.

Caution: Combining too many terms together with AND increases the chances of creating a null set.

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 056, PARENT = 047, LEVEL = 008, NO. SUBORDINATE RECDs = 001

With the FIND OR command you may use up to five terms. You may use fewer than five, but not less than two.

This command is equivalent to the sequence:

FIND TERM1 OR TERM2

FIND /P/ OR TERM3

FIND /P/ OR TERM4

FIND /P/ OR TERM5

but FIND OR is quicker.

..CONTINUE

1. (057) More about FIND OR
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 057, PARENT = 056, LEVEL = 009, NO. SUBORDINATE RECDs = 000

Caution: Combining too many terms together with OR increases the chances of creating a set too big for the computer storage allocated (200 records). If this happens, the command will not be fully carried out. It will stop after 200 records have been retrieved; you will not know how many records you may have missed.

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 058, PARENT = 048, LEVEL = 008, NO. SUBORDINATE RECDs = 000

If you do not want to review any more, you say NO and you are ready to resume the search.

If you have been doing a lot of browsing, it is easy to lose track of what you have been trying to do. HIST helps you review your work to help you plan what you are going to do next.

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 059, PARENT = 029, LEVEL = 009, NO. SUBORDINATE RECDs = 002

If a term you looked up in the dictionary is not found, it shows up in the list. However, instead of a record count, you see the legend NOT IN DICT.

If you want to see other nearby words, you can use either of the commands PAGE DICT UP (or U) or PAGE DICT DOWN (or D).

1. (062) PAGE DICT UP
2. (063) PAGE DICT DOWN
3. (001) RETURN TO ROOT
4. (...) QUIT

*** 060, PARENT = 006, LEVEL = 004, NO. SUBORDINATE RECDs = 001

The dictionary is an alphabetically ordered list of all words used in index or data records EXCEPT certain so-called STOP words, which are deemed to convey no information. These include common words, such as AND, OF, OR, and any string of symbols beginning with a numeral, such as 1981 or 5B.

..CONTINUE

1. (061) More about the dictionary
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 061, PARENT = 060, LEVEL = 005, NO. SUBORDINATE RECDs = 000

With each word is a count of the number of records in which the word occurs--index or data. You can use the dictionary to help select search words and to see how often various words are used in the database. Searching the dictionary is described under SEARCHING - BASIC COMMANDS.

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 062, PARENT = 059, LEVEL = 010, NO. SUBORDINATE RECDs = 001

PAGE DICT UP or PAGE DICT U moves up five words in the dictionary. You see the five words preceding the top one on the previous list.

You cannot use PAGE DICT U unless you have previously used DICT TERM in the current search.

It is possible to PAGE yourself right off the end of the dictionary.

..CONTINUE

1. (064) Reaching the end of the dictionary
2. (001) RETURN TO ROOT
3. (...) QUIT

*** 063, PARENT = 059, LEVEL = 010, NO. SUBORDINATE RECDs = 002

PAGE DICT DOWN or PAGE DICT D moves down five words in alphabetical order in the dictionary. You will see the five words following the last word on the previous list.

You cannot use PAGE DICT D unless you have previously used a DICT TERM command.

It is possible to PAGE off the end of the dictionary.

..CONTINUE

1. (065) Reaching the end of the dictionary
2. (001) RETURN TO ROOT
3. (...) QUIT

Date: 1983 July 15

*** 064, PARENT = 062, LEVEL = 011, NO. SUBORDINATE RECDs = 000

You can repeat PAGE DICT U as often as you want; each time you go up five more words.

If you run off the end of the dictionary, you will see the notation

END OF DICTIONARY

instead of the next word. To continue browsing, just PAGE in the opposite direction, i.e., PAGE DICT DOWN to see the top five words in the dictionary.

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 065, PARENT = 063, LEVEL = 011, NO. SUBORDINATE RECDs = 000

You can repeat PAGE DICT D as often as you want and you will move down five words each time. If you reach the end of the dictionary, you will see the legend

END OF DICTIONARY

instead of a word. To continue browsing, PAGE DICT in the opposite direction, i.e., PAGE DICT U. You will go back to the page with the last five words on the dictionary.

1. (001) RETURN TO ROOT
2. (...) QUIT

*** 066, PARENT = 005, LEVEL = 004, NO. SUBORDINATE RECDs = 000

You cannot quit the search while in HELP. If you want to quit, return to the search first and then use the QUIT option or the QUIT command.

This is now the end of the material on HELP. There is no subordinate to this record. Your only choices are to return to the root of HELP to follow a different path or return to your main search.

1. (001) RETURN TO ROOT
2. (...) QUIT

APPENDIX B. Test Questions

The same set of questions was given to each subject in the same order. The question number (sequentially numbered beginning with 1) is found in the last three digits of the seven-digit code. The third digit of this code is the rating of question difficulty.

The first digit of the search number identifies the session in which the question was used; the second digit identifies the search within the session.

QUESTION: 3821001 -----SEARCH NO.: 11 -----USER PASSWORD:

If you were buying sugar on the commodities market in January 1981, what price would you have paid?

Enter below the numbers of any records that comprise part of the answer.

QUESTION: 9301002 -----SEARCH NO.: 12 -----USER PASSWORD:

Inflation is a worldwide problem, but in comparison with Israel almost any western nation might feel fortunate. During the autumn of 1980 what predictions were made for Israel's 1980 annual inflation rate?

Enter below the numbers of any records that comprise part of the answer.

QUESTION: 1612003 -----SEARCH NO.: 13 -----USER PASSWORD:

Your government bureau is trying to create some economic good news and is preparing some facts and figures on foreign trade. Can you identify a country that is buying several thousand tonnes of railroad track from Britain?

Enter below the numbers of any records that comprise part of the answer.

QUESTION: 8462004 -----SEARCH NO.: 14 -----USER PASSWORD:

An intelligence network has reported that Jordan has recently acquired some American-made tanks. How has she gotten them?

Enter below the numbers of any records that comprise part of the answer.

QUESTION: 6331005 -----SEARCH NO.: 21 -----USER PASSWORD:

You are gathering data on U.S. investments in Britain in 1980. Which American firm bought Fodens, the British truck maker?

Enter below the numbers of any records that comprise part of the answer.

QUESTION : 5711006 -----SEARCH NO.: 22 -----USER PASSWORD:

You are researching the football scene in 1980 for an ITV special programme later this year. What world cup matches did Kevin Keegan miss in 1980?

Enter below the numbers of any records that comprise part of the answer.

QUESTION: 1822007 -----SEARCH NO.: 23 -----USER PASSWORD:

In the early days of the Gulf War, American observers believed that Iraq had specific territorial objectives. What were these?

Enter below the numbers of any records that comprise part of the answer.

QUESTION: 5713008 -----SEARCH NO.: 24 -----USER PASSWORD:

You are gathering data for an article reviewing the mining industry in Britain. Find two incidents of industrial action by miners during the autumn of 1980 as well as the wage claims of miners under negotiation in November 1980. Also include the number of pits the NCB plans to close in 1981.

Enter below the numbers of any records that comprise part of the answer.

QUESTION 4831009 -----SEARCH NO. 31 -----USER PASSWORD:

Although BSC has apparently insurmountable financial difficulties, its River Don works has won some new orders. What were they worth?

Enter below the numbers or any records that comprise part of the answer.

QUESTION: 8492010 -----SEARCH NO: 32 -----USER PASSWORD:

You are preparing some summary data on the Iranian government as introductory lecture material, giving the names and offices of the present power elite. What is the name of the political party now in power in Iran, and who is the party leader?

Enter below the numbers of any records that comprise part of the answer.

QUESTION: 7382011 -----SEARCH NO.: 33 -----USER PASSWORD:

Your sports editor has decided to publish a list of highly paid British soccer players. Whose name is likely to appear at the top of such a list?

Enter below the numbers of any records that comprise part of the answer.

QUESTION: 9453012 -----SEARCH NO.: 34 -----USER PASSWORD:

The Morning Telegraph (Sheffield) is reviewing the economic consequences of the recession on Sheffield and Rotherham in 1980. Its writers need a list of plant closings in these two cities. Name two firms to be included.

Enter below the numbers of any records that comprise part of the answer.

QUESTION: 2121013 -----SEARCH NO.: 41 -----USER PASSWORD:

Although most of the strikes in 1980 were called over wages and economic benefits, there was one called over the refusal of permission for a worker to play in a pub darts match. where was it?

Enter below the numbers of any records that comprise part of the answer.

QUESTION: 7322014 -----SEARCH NO.: 42 -----USER PASSWORD:

You are gathering intelligence for a western nation on Soviet involvement in the Gulf War. You've been advised that Russian building supplies are being sent to Iraq. How are they getting there?

Enter below the numbers of any records that comprise part of the answer.

QUESTION: 7353015 -----SEARCH NO.: 43 -----USER PASSWORD:

The financial editor has asked you to prepare a review article of the trials, tribulations and successes of British Leyland in the past few months, so you are gathering data to document it. You need to include: a plant closure, a new product, a cooperative project with BR, a trading loss report and a 1980 wage offer.

Enter below the numbers of any records that comprise part of the answer.

Date: 1983 July 15

QUESTION: 6973016 -----SEARCH NO.: 44 -----USER PASSWORD:

You are preparing an article on the impact of the Gulf War on Iran's oil industry. Some of the data you need to include are the following: the name of the Oil Minister, estimates of oil production at the outbreak of the war, information on exports, and pricing policies for personal consumption within Iran.

Enter below the numbers of any records that comprise part of the answer.

QUESTION: 8201017 -----SEARCH NO.: 51 -----USER PASSWORD:

You are charting the retail price index for 1980. What was it in August?

Enter below the numbers of any records that comprise part of the answer.

QUESTION : 7312018 -----SEARCH NO.: 52 -----USER PASSWORD:

You are interested in diversifying your investment portfolio to include some commercial stock such as supermarkets. Looking at the performance of such stocks in 1980 which supermarket chain had the best record of share price increases in the last quarter of 1980?

Enter below the numbers of any records that comprise part of the answer.

QUESTION: 9253019 -----SEARCH NO.: 53 -----USER PASSWORD:

British Rail has been asked to prove that it does not discriminate against the employment and promotion of women. Find one piece of evidence BR could bring to its defence.

Enter below the numbers of any records that comprise part of the answer.

QUESTION: 4863020 -----SEARCH NO.: 54 -----USER PASSWORD:

If you were convening an international conference of foreign ministers, who would be representing Iraq and Israel?

Enter below the numbers of any records that comprise part of the answer.

APPENDIX C. Training Handout

The document on the next five pages comprises the complete set of notes given to subjects. Those assigned Language 1 were given only pages 1 and 2. Those assigned Language 2 were given pages 1-4. Those assigned Language 3 were given all five pages.

L U R E S System Summary

L U R E S Search System A Brief Summary

DATABASE STRUCTURE

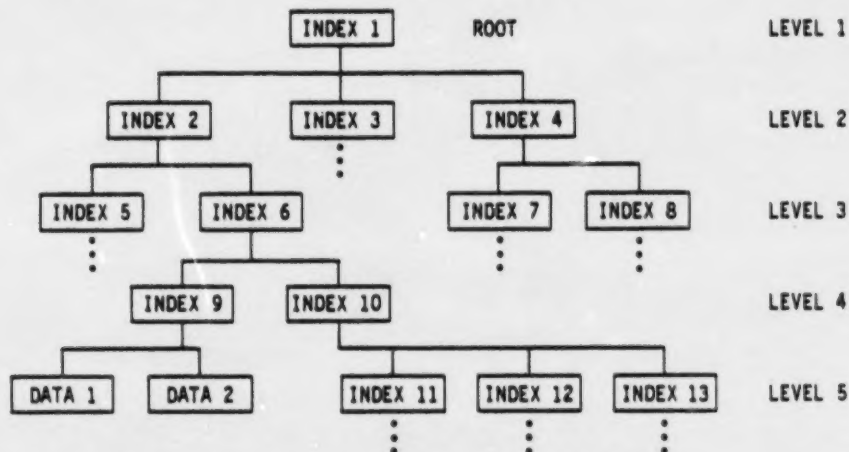
Database: a collection of news items arranged by subject, hierarchical in structure, i.e., subjects may be repeatedly subdivided for as many as eight levels.

Index Record: a record in the database containing only subject headings.

Data Record: a record of the database containing substantive information.
All the data records are short news items.

Index records have subordinate records, i.e., those describing further subdivisions of the subject or those containing data. Data records do not have subordinates. A record that is subordinate to another record has the latter as its parent.

All records have parents except the first index record which is called the index root. The index root is considered to be at level 1, its subordinates at level 2, and their subordinates at level 3, etc.



INDEX 1 is the parent of INDEX 2,3 & 4. INDEX 2,3 & 4 are offsprings of INDEX 1.

INDEX 5 & 6 are siblings (same parent). INDEX 6 & 7 are not siblings.

The Hierarchy of Index and Data Records

L U R E S System Summary

LOG IN

The computer asks you to LOG IN. Key in your four-digit searcher password. A few more preliminary questions will ask for the search number, which is two digits, and the question number--the seven-digit number on your question card. Then you will see the index root record and the input cue :: displayed and you are ready to search.

INPUT CUE ::

Following the display of the index root record and after any subsequent input of an option command by you, the symbol :: cues you to respond with a selection of an option.

SELECTING AN OPTION

When any record, index, or data is displayed, a list of options is provided for what can be done next. If the record is an index record, these include: a) getting any of the subordinates of the current record, b) returning to the index root record (except when the root record is the one displayed) and, c) quitting the search. Selection of QUIT is followed by a question asking if you really intend to quit--a check against inadvertant quitting. To select an option, key in the number of the statement of the option and then press the RETURN key.

CONCLUDING A SEARCH

When you have completed a search, that is, one of your search questions has been answered to your satisfaction or you have decided to quit, select the QUIT option following the input cue ::. You will then move on to two evaluation questions to be answered in the same manner as selecting an option in the search. After completing these, the terminal will display a message stating that the search has been completed and instructing you how to begin another, if you wish. To initiate another search, following the display of the symbol > on the screen you key in the following sequence of instructions to the computer:

```
QUIT (R) [(R) denotes RETURN key]
CLOSE ALL (R)
BASICV (R)
OLD SEARCH (R)
EXECUTE (R)
```

Then you will see the message asking you to log in and you can proceed as you did for the first question as described above under the heading LOG IN.

CORRECTIONS

To correct your selection before hitting the return key press the ? and then enter the correct selection.

ALWAYS WAIT FOR THE :: BEFORE ENTERING A SELECTION.

ALWAYS EXECUTE YOUR CHOICE BY PRESSING THE RETURN KEY.

ALWAYS END YOUR SEARCH WITH THE QUIT OPTION.

L U R E S System Summary

SEARCHING BY COMMAND

Whenever the :: cue appears, you may select an option by number or enter a command. The commands are:

PAGE R, L, U, D. These are four variations of the same command. Each requests display of another "page" of information, or another record. These four enable you to view other records within the index-data hierarchy. Following the display of any record you may command:

PAGE DOWN (D) which is exactly equivalent to selecting option 1 for an index record. It gets the first subordinate or offspring of the current record. It has no meaning for a data record.

PAGE UP (U) gets the parent of the current record. It has no meaning if the current record is the index root.

PAGE RIGHT (R) gets the next sibling of the current record.

PAGE LEFT (L) gets the previous sibling of the current record.

Both PAGE R and PAGE L operate in a circular manner. PAGE R after the last sibling gets the first; PAGE L after the first sibling gets the last.

FIND "term". This command finds all records in the database, both index and data records that contain the "term" you have selected as part of the command. The term may be any word, but one word only. The result is a statement of how many records contain this term and a display of the first of the set of records containing it. Only dictionary terms will successfully create a set, that is terms used in the database either in the index or data records. (An explanation of the dictionary appears below under the heading DICT "term.") If your term is not in the dictionary you will create a null set and will be so informed. The order of appearance of records in a set is from the lowest to the highest level in the hierarchy, that is those nearest the root index appear first.

PAGE SET U, D. These commands enable you to view the records of a set created by a FIND command, one at a time, beyond the first one. PAGE SET DOWN (D) gets the next record in level order, and PAGE SET UP (U) gets the previous record. PAGE SET, like PAGE L or R, operates in a circular manner, so PAGE SET D following the last record in a set gets the first one and PAGE SET U following the first record gets the last one.

DICT "term". This command causes your term to be looked up in the database dictionary. The term must be a single word. If it is present, it and its frequency are displayed along with the two words preceding and following it and their frequencies for a total of five words in the display with your "term" in the center. If the "term" is not in the dictionary, the note ..NOT IN DICT.. appears instead of a frequency statement. The dictionary is an alphabetical listing of all the words in either data or index records in the database, including a count of the number of records in which the word occurs.

Omitted from the dictionary are common words such as "and," "or," "the". . . and any term beginning with a numeral such as 1981 or 2M.

PAGE DICT U, D. These commands enable you to page through the dictionary. PAGE DICT DOWN (D) creates a display of the next five words, following the last word displayed after the most recent DICT or PAGE DICT command. PAGE DICT UP (U) gets the preceding five words. These commands are not circular. If you reach the end of the dictionary, the note ...END OF DICTIONARY... appears. Counter this by moving a page in the opposite direction.

QUIT. This command has exactly the same meaning as selecting the QUIT option.

CHOICE-OF-PATH SEARCHING. You can at any time select an option by number, freely interspersing these choices with commands.

HELP. This command gives you access to a file of explanatory information about the database, the search system, and the HELP file. This file is organized like the database and is searched only by the choice-of-path method. The cue for your option selection in the HELP file is ::. You may request HELP at any time during your search, and when you select the option return to search" you will go back to the search exactly where you left it. You cannot QUIT from HELP.

CORRECTIONS

To correct a spelling or keying error, you have 2 options:

To correct one letter or symbol, you backspace by keying " and replace the incorrect letter with the correct one.

To correct an entire line or command, you key in ? which deletes the line; then you replace it with the correct complete line. Do not use the backspace key; it has no meaning for this system.

ALWAYS WAIT FOR THE :: BEFORE ENTERING A COMMAND.

ALWAYS EXECUTE YOUR COMMAND BY PRESSING THE RETURN KEY.

ALWAYS END A SEARCH WITH A QUIT..

Date: 1983 July 15

L U R E S System Summary

ADVANCED COMMANDS

FIND ROOT :. This command searches for all records that contain words beginning with the string of letters provided as a word or term ROOT. It is essential to leave a space between the end of the root and the character ":" or the computer thinks of your terms as being ROOT:--all one word and it looks this up in the dictionary. This enables you to collect all the variations of a term in one set. For example, FIND BRIT : would create a set of all records containing the words BRITAIN, BRITAINS, BRITISH, and BRITON.

FIND TERM1 AND TERM2
FIND TERM1 OR TERM2
FIND TERM1 NOT TERM2

This command enables you to search for records that have both terms (AND), either or both terms (OR), or the first but not the second of the two terms given.

FIND /P/ AND TERM2
FIND /P/ OR TERM2
FIND /P/ NOT TERM2

This command enables you to modify an existing set. /P/ refers to the previous set. FIND UK followed by FIND /P/ or BRITAIN is equivalent to the single command FIND UK OR BRITAIN. You may continue to modify the existing set as often as you like.

FIND AND TERM1 TERM2 . . . TERM5. This command enables you to form a set based on as many as five terms (but a minimum of two) without having to use repetitive FIND /P/ commands. All the terms are treated as if connected by AND.

FIND OR TERM1 TERM2 . . . TERM5. This command enables you to form a set based on as many as five terms (but a minimum of two) without having to use repetitive FIND /P/ commands. All the terms are treated as if connected by OR.

HIST. This command enables you to review your search. It lists the set-creating FIND commands that you have executed thus far (not any that have been rejected), up to 18 at a time, and if you have done more than 18 you can see another page of up to 18, etc. HIST can be used at any time during a search. When you have finished with it, you will return to your search exactly where you left it.

ALWAYS WAIT FOR THE :: BEFORE ENTERING A COMMAND.

ALWAYS EXECUTE A COMMAND BY PRESSING THE RETURN KEY.

ALWAYS END A SEARCH WITH QUIT.

APPENDIX D. End of Search Questions

These questions were asked automatically at the conclusion of each individual search, and the data recorded as part of the search.

1. Please evaluate how well you feel you conducted this search, independently of any particular records retrieved or results attained. Use a scale of 5 where:

1 = Very poorly
2 = Poorly
3 = Neutral
4 = Well
5 = Very well

2. Please evaluate how well you feel the outcome of this search, i.e., the records retrieved, answers the question, independently of how the search was performed. Use a scale of 5 where:

1 = Very poorly
2 = Poorly
3 = Neutral
4 = Well
5 = Very well

APPENDIX E. End of Session Questions

Because the languages were different, there are three different evaluation questionnaires, similarly organized, but asking different detailed questions about the languages. These were asked at the end of each session of four searches. The language number that appears on the questionnaire did not appear when the subjects saw these forms. They were supposed to be unaware that there was more than one language.

Date: 1983 July 15

Language 1

System Evaluation Questionnaire

The questions below concern your evaluation of the system you have been using from the point of view of how good a tool it is for searching a database. Please answer only in regard to THIS session, not to any searches completed in earlier sessions.

Your answers should reflect a composite attitude toward all the questions you worked on during the session just completed.

Each line of the questionnaire consists of the name of a system element, such as Return to root, and a set of columns in which you can tick an answer. For each line, indicate the contribution you think the element makes to a search. Answers may range from Hinders Very Much to Helps Very Much with several variations in between. The center column is headed "NEUTRAL" and it indicates neither a positive nor a negative feeling toward that element.

For example, if a system feature were TERMINAL KEYBOARD, you might feel it hindered you a little, and you will find an answer--the third column--that expresses this thought. If you had felt the keyboard was very much help to you, you might have ticked the right-most column. If you felt it neither helped nor hindered, you would have ticked the middle column.

System Element	HINDERS			NEUTRAL			HELPS		
	Very much	Some	Little	Little	Some	Very much			
SEARCHING									
Choice of subordinate.....									
Return to root.....									
Quit.....									
DATABASE									
Hierarchical data/index structure.....									
The system as a whole.....									

Date: 1983 July 15

Language 2

System Evaluation Questionnaire

The questions below concern your evaluation of the system you have been using from the point of view of how good a tool it is for searching a database. Please answer only in regard to THIS session, not to any searches completed in earlier sessions.

Your answers should reflect a composite attitude toward all the questions you worked on during the session just completed.

Each line of the questionnaire consists of the name of a system element, such as Return to root, and a set of columns in which you can tick an answer. For each line, indicate the contribution you think the element makes to a search. Answers may range from Hinders Very Much to Helps Very Much with several variations in between. The center column is headed "neutral" and it indicates neither a positive nor a negative feeling toward that element.

For example, if a system feature were TERMINAL KEYBOARD, you might feel it hindered you a little, and you will find an answer--the third column--that expresses this thought. If you had felt the keyboard was very much help to you, you might have ticked the right-most column. If you felt it neither helped nor hindered, you would have ticked the middle column.

System Element	HINDERS			NEUTRAL			HELPS		
	Very	Some	Little	Little	Some	Very	Very	Some	much
SEARCHING									
Choice-of-path method of searching.....									
PAGE command.....									
FIND command.....									
PAGE SET command.....									
DICT command.....									
PAGE DICT command.....									
HELP command.....									
QUIT command.....									
DATABASE									
Hierarchical data/index structure.....									
Dictionary.....									
The system as a whole.....									

Language 1

System Evaluation Questionnaire

The questions below concern your evaluation of the system you have been using from the point of view of how good a tool it is for searching a database. Please answer only in regard to THIS session, not to any searches completed in earlier sessions.

Your answers should reflect a composite attitude toward all the questions you worked on during the session just completed.

Each line of the questionnaire consists of the name of a system element, such as Return to root, and a set of columns in which you can tick an answer. For each line, indicate the contribution you think the element makes to a search. Answers may range from Hinders Very Much to Helps Very Much with several variations in between. The center column is headed "NEUTRAL" and it indicates neither a positive nor a negative feeling toward that element.

For example, if a system feature were TERMINAL KEYBOARD, you might feel it hindered you a little, and you will find an answer--the third column--that expresses this thought. If you had felt the keyboard was very much help to you, you might have ticked the right-most column. If you felt it neither helped nor hindered, you would have ticked the middle column.

<u>System Element</u>	<u>HINDERS</u> Very much Some Little	<u>NEUTRAL</u> Little Some much	<u>HELPS</u> Very much
<u>SEARCHING</u>			
Choice of subordinate.....	_____	_____	_____
Return to root.....	_____	_____	_____
Quit.....	_____	_____	_____
<u>DATABASE</u>			
Hierarchical data/index structure.....	_____	_____	_____
The system as a whole.....	_____	_____	_____